California State University
SAN MARCOS

Course Project:

Campus Parking Management System

Team:

Kevin Truebe

Chris Banci

CIS 341

System Analysis and Design

Prof. Margo Lopez

Cal State San Marcos

May 15, 2017

[Table of Contents]

[Overview]

The current parking management system is outdated and needs improvement. It is being hosted by a cloud vendor that is not keeping up with innovative technologies. To change this, the new director of parking services is requesting a new improved system that will incorporate automation into the existing system, such as automated ticket payment and license plate recognition. The minimum requirements for this new system are license plate recognition and a mobile payment interface.

In regards to this, there are few things that can be improved in the current parking management system: Firstly, the current system does not have a way for parking enforcement to automatically identify vehicles with parking violations such as having no permit or expired parking times. Secondly, the current system does not offer a way for students to purchase parking tickets automatically. Thirdly, the current system does not provide a way for students to view available parking around campus.

The proposed improved system will include the following: Firstly, the migration to Amazon Web Services (AWS) as they provide the necessary computing power, database storage and functionality to scale and grow the system. Secondly, an automated way to pay for parking through the use of RFID technology. Thirdly, the ability for parking enforcement to recognize license plates. Lastly, the ability way to track available parking around campus through counter displays and a mobile application.

[System Planning]

By analyzing the issues present in the current system as discussed in the overview, we have established a set of new features that will be of great benefit in the existing system. Furthermore, we have taken into account the current shortcomings of the system and hypothesized solutions to test going forward.

Business Process

Process	Objective	Input	Output	Tasks	Outcomes
Vehicle Management process	Maintain a database of all students that park on campus	Vehicle information of students and faculty such as license plate and VIN.	Ability to access vehicle information within the system.	Add vehicle information to database	Complete and accurate information of all vehicles parked on campus used for parking enforcement.
Payment Processing	To process parking payment for the automated parking solution.	Request for parking. Credit card information.	Parking time is received	Issue RFID tags to students.	Time and convenience is saved when paying for parking.
Vendor Migration Process	To migrate to a vendor that is up-to-date.	Data from current system	System has new data	Migrate old data to new system.	Campus parking uses innovative technology
Parking Enforcement Process	To use license plate technology to assist in parking enforcement	License plate data from camera feed	Citation logged and issued out.	Equip parking enforcement with LPR equipment.	Easier for parking enforcement to easily identify vehicles with parking violations and issue citations.

Use Case

This use case explains the process of how a parking spot updates its occupancy status.

Name	Parking Lot Occupancy Process
Actor	Student
Successful Completion	Student parks vehicle in parking spot Parking sensor detects vehicle
Alternative	Student does not park vehicle in parking spot. Parking sensor does not detect occupancy.
Precondition	Student must have a vehicle
Post condition	Available parking display is updated.
Assumptions	None

This use case explains how parking is validated when a student preloads their RFID tag.

Name	Parking Validation Process for Pre-loaded tags
Actor	Student
Successful Completion	Student has already preloaded their tag. Student passes through RFID sensor System verifies the tags on the car. System does not need to track this car after check-in.
Alternative	Student does not have preloaded tags System notifies Parking Enforcement
Precondition	Student has agreed online to pay via tags and has received tags in mail. Student has already pre-loaded their parking account with funds.
Post condition	Amount is deducted from preloaded account upon leaving campus.
Assumptions	None

This is use case explains how temporary tags are received if one forgets their RFID tags.

Name	Backup Parking Process (Visitor or forgot tags)
Actor	Student
Successful Completion	Student has valid Credit or Debit Card ready for machine Student prints temporary tags
Alternative	Student does not use temporary tags System notifies Parking Enforcement
Precondition	Student has agreed to on-screen terms at terminal about parking rules as well as price of parking
Post condition	Credit Card is charged when swiped at parking terminal
Assumptions	None

This is use case that explains how parking citations are automatically issued out.

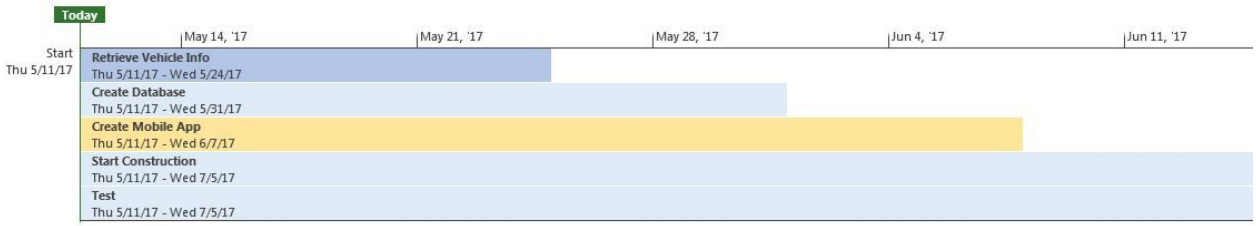
Name	Parking Citation Issuance Process
Actor	Parking Enforcement
Successful Completion	LPR runs license plate of student's vehicle in database Student does not have valid parking permit or time allotted is expired. Student is issued citation and notified about the citation.
Alternative	Student has valid parking permit or time allotted. Student is not issued citation.
Precondition	Student has agreed to on-screen terms at terminal about parking rules as well as price of parking
Post condition	Student pays for parking citation.
Assumptions	None

Project Plan

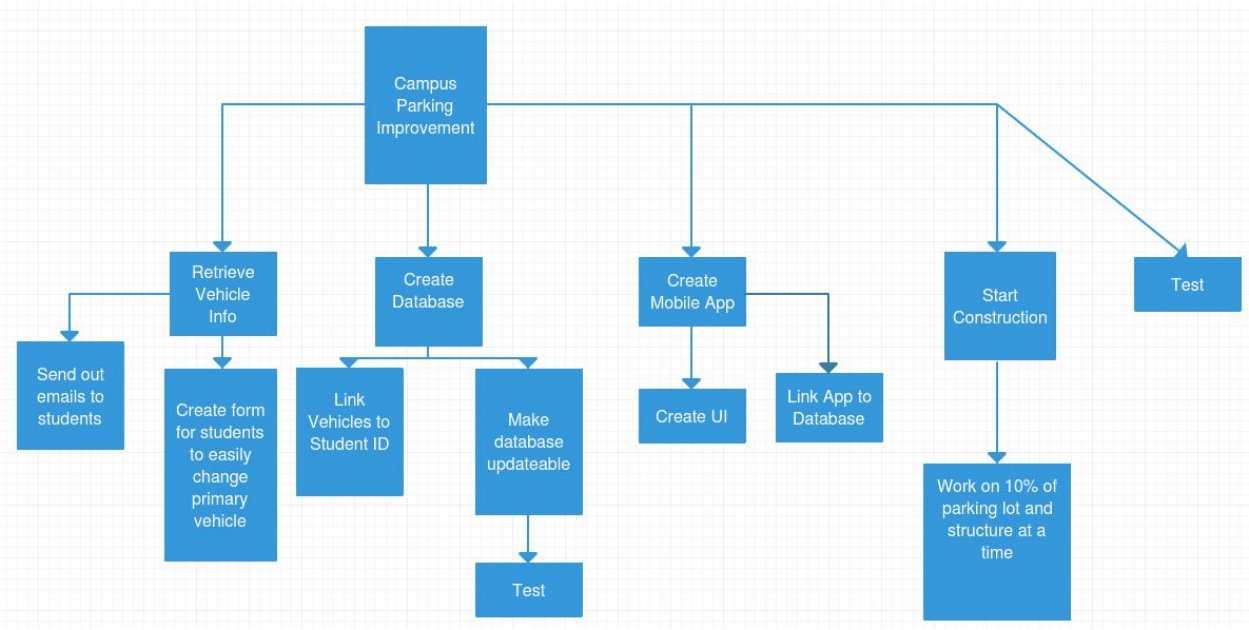
It is estimated that the project will take 7 full weeks to complete. The project will be deployed during the summer when students are on break. The tasks in our project plan include retrieving vehicle information, creating a database, creating the mobile application, starting the construction, and continually testing until everything works. Because we are managing the project using the agile method, each task can be started at the same time.

Task Name	Duration	Start	Finish
Retrieve Vehicle Information	**10 days**	**Thu 5/11/17**	**Wed 5/24/17**
Send e-mail to all returning and new students about the new parking plan, ask for primary and secondary vehicle license plates	7 days	Thu 5/11/17	Fri 5/19/17
Create form for students to easily change their primary vehicle and be updated in the database	3 days	Mon 5/22/17	Wed 5/24/17
At this point, hopefully most of the students will have submitted their data. Link will stay open for all students	0 days	Wed 5/24/17	Wed 5/24/17
Create the Database	**15 days**	**Thu 5/11/17**	**Wed 5/31/17**
Link the vehicle license plate with each Student ID	10 days	Thu 5/11/17	Wed 5/24/17
Make the database updatable for students	1 day	Thu 5/25/17	Thu 5/25/17
Test the Database	4 days	Fri 5/26/17	Wed 5/31/17
At this point the database will be ready to go and tested	0 days	Wed 5/31/17	Wed 5/31/17
Create the Mobile Application	**20 days**	**Thu 5/11/17**	**Wed 6/7/17**
Create the user interface	10 days	Thu 5/11/17	Wed 5/24/17
Link the application to the database	10 days	Thu 5/25/17	Wed 6/7/17
The application should the available spaces on each level of the parking garage or each section of the parking lot	0 days	Thu 5/11/17	Thu 5/11/17
Start Construction	**40 days**	**Thu 5/11/17**	**Wed 7/5/17**
Work on 10% of the parking lot and parkin structure at a time to allow summer students space to park	40 days	Thu 5/11/17	Wed 7/5/17
Test	**40 days**	**Thu 5/11/17**	**Wed 7/5/17**
Test the new system	40 days	Thu 5/11/17	Wed 7/5/17

Gantt Chart



Work Breakdown Structure



[System Analysis]

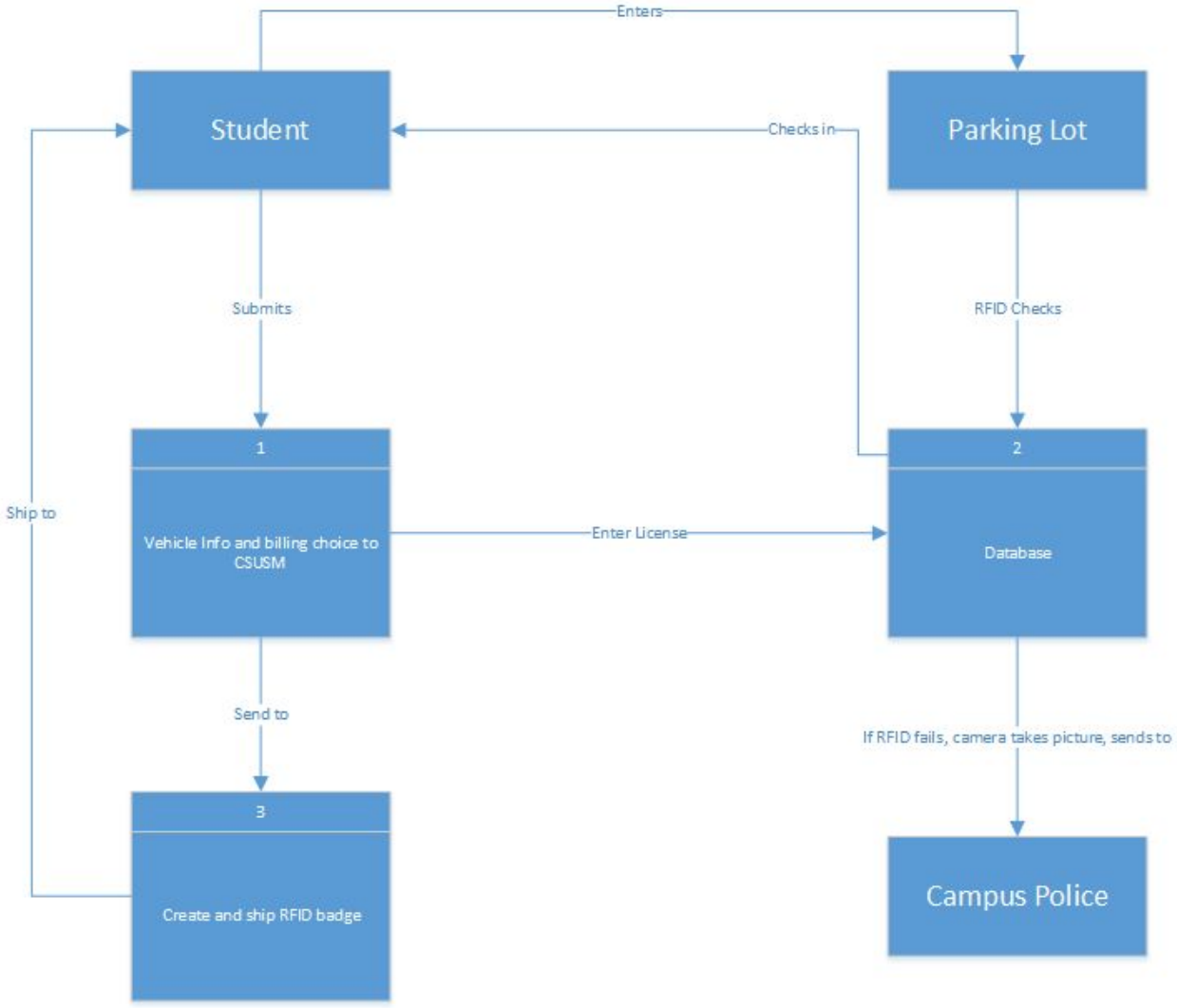
System Requirements

System Features	Priority	Current Custom Solution	Requirements
License Plate Recognition (LPR)	Must have	Cameras exist for security, but not for license plates.	The camera system should identify license plates of vehicles and run them through a database. The camera system should be installed at every parking lot.
Backup Tag System	Nice to have	None	The backup tag system should provide students with a temporary pass if they forget their RFID tags.
Available Parking Displays	Must have	None	These displays should show available parking for each parking lot.
Mobile application	Must have	A CSUSM mobile application exists, but it does not provide much functionality.	The mobile application should provide a way to manage parking payments and view the exact locations of available parking across campus.
Automated parking payment	Must have	Four ways to pay for parking exists, but none are automatic.	This should be implemented with the use of RFID technology. Sensors in each parking lot will need to be installed which will interface with the RFID tags. Students will have the option to pay on the go or preloaded.
Migration to Amazon Web Services (AWS)	Must have	Current cloud vendor is outdated.	The new vendor should be keeping with new innovations.

Software Requirements

Requirements	Description
License plate recognition software	This is needed for cameras to identify license plates.
Parking sensor software	This is needed to interface with parking sensors so that information on displays and mobile application can be updated.
Automated ticket payment software	This is needed to interface with RFID tags for automated ticket payment.
Camera software	This is needed to interface with the cameras installed around campus
AWS software	This is needed to manage the cloud services.

Data Flow



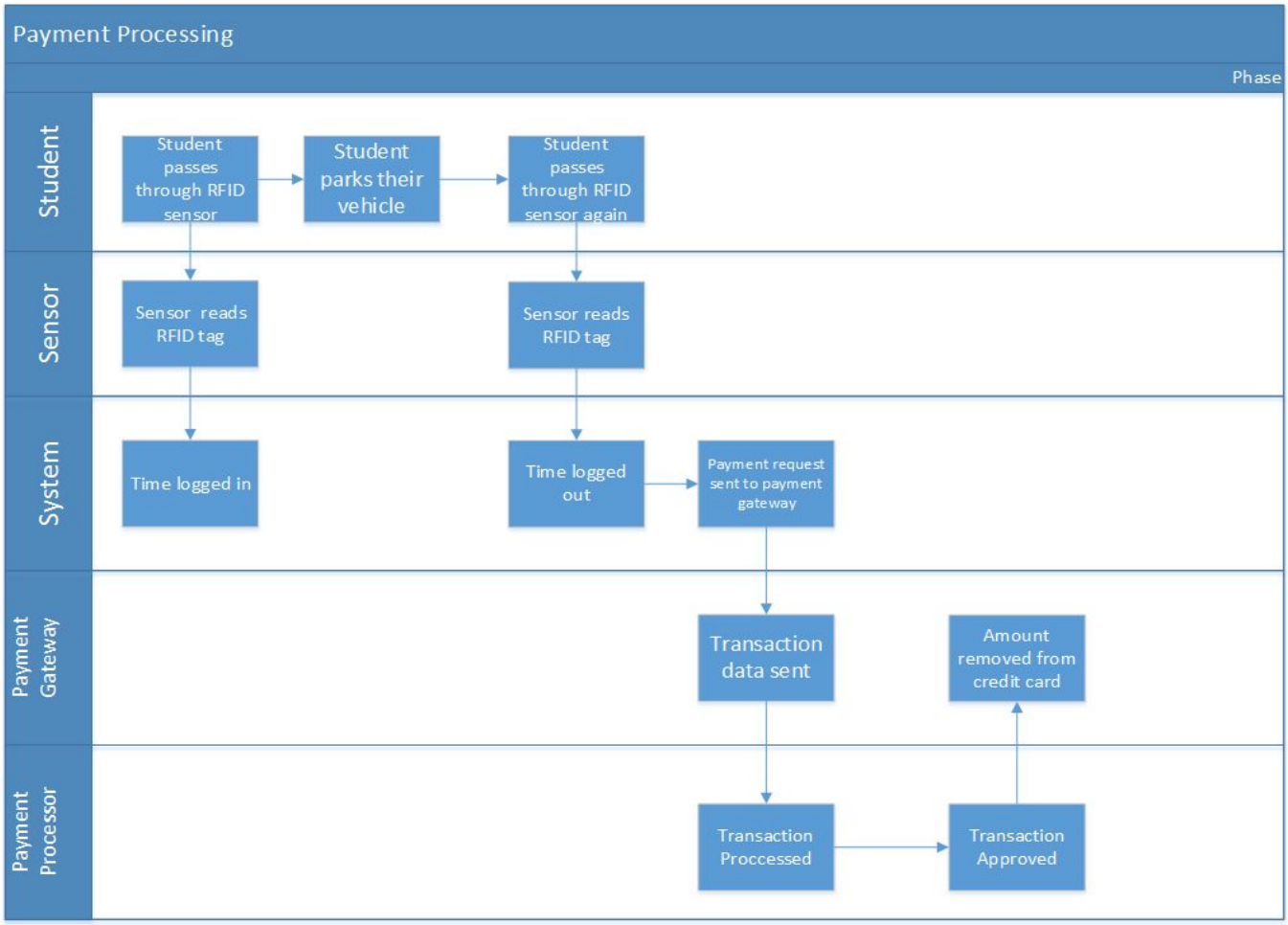
This is our data flow diagram. The student will submit their primary and alternate vehicle to CSUSM to be entered into the database. The License will be entered into the datasebe and a third party will create and ship the RFID badge to the student. The student may then choose to preload, pay as you go, or pay for the full semester outright with their credit card. Now, once the student enters the parking lot, the RFID will check them in. If they have no RFID tag, the camera system will detect the licence plate, take a picture, and alert campus police.

Security Constraints

Constraint	Description
PCI Compliance	Needed to keep the system and customer information secure when accepting student payment card information.
Restricted Access	Sensitive information needs to be secure. To achieve, access to sensitive information will need to be restricted to specific users. Students will only have access to their information like vehicle, address, and credit card.
Encryption	Credit card information needs to be encrypted instead of being stored in plaintext.

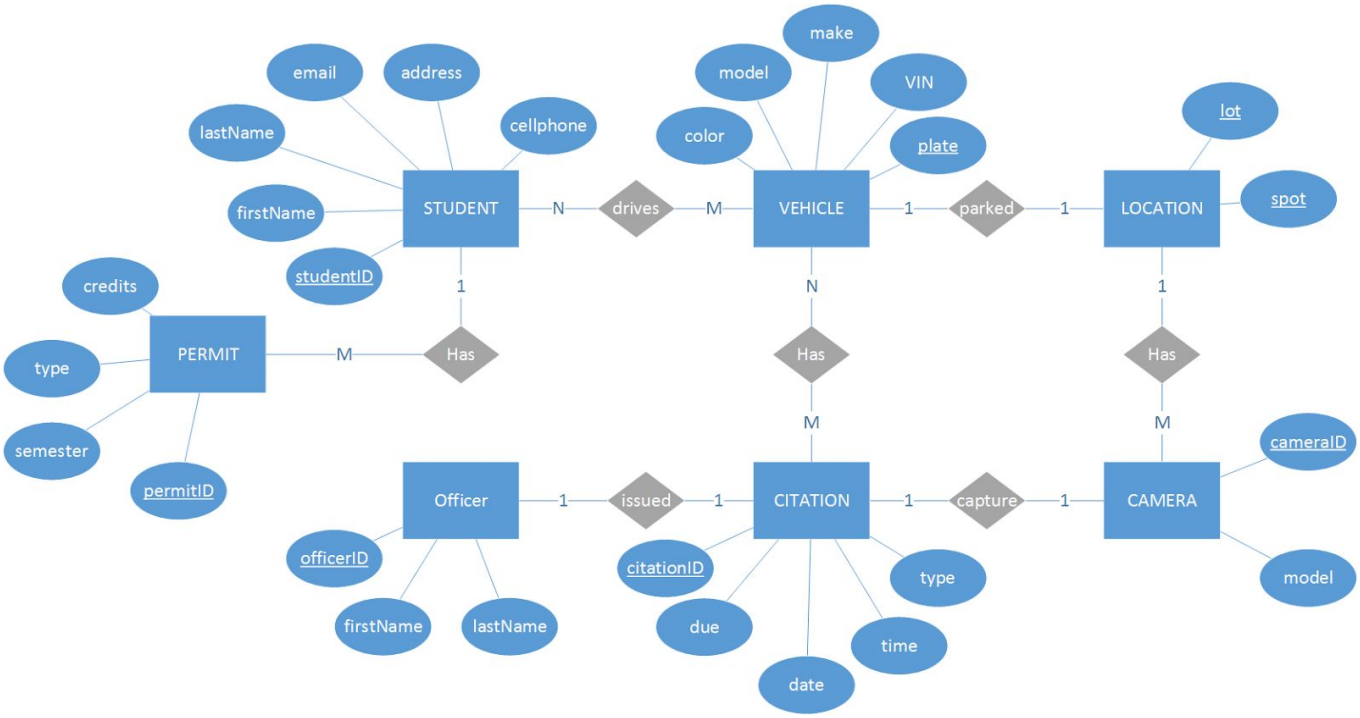
Business Process Swim Chart

This swim chart explains the business process of payment processing.



[System Design]

Database Design



The diagram above is the layout of our database. The type of database that will be used for our system will be a relational type database and it will be hosted on Amazon Web Services. Since some of the attributes of each entity contain sensitive information such as credit card information and student address, they will need to be encrypted for security purposes. Amazon Web Services provides this all. The database will be managed through a database client like MYSQL.

Below are the tables for our database:

- PERMIT (permitID, credits, type, semester, studentID*)
- STUDENT (studentID, firstName, lastName, email, address, cellphone)
- STUDENT_VEHICLE(studentID*, plate*)
- VEHICLE (plate, vin, color, model, make)
- LOCATION (lot, spot, plate*)
- CAMERA (cameraID, model, (lot, spot)*)
- CITATION (citationID, due, date, time, type, cameraID*, officerID*)
- CITATION_VEHICLE(citationID*, plate*)
- OFFICER (officerID, firstName, lastName)

The primary keys of each table are the attributes underlined.
The foreign keys of each table are identified with an asterisk*.

Development Process

For project management, we will be using the agile process. We believe that using this team based approach will help us complete the project on time and efficiently. Using the agile process, we can make sure that there is a working model by the time students return for the fall semester, even with setbacks. With this process, the focus is based on the student, or our customer, meaning we will create a better product for our users.

We will manage a team of developers with daily scrum meetings. These meetings will help our team stay in sync, build trust, maintain visibility of the project for all members of the group and help make corrections for the sprint. These meetings will help our team stay focused and understand each other's work as well as help clear roadblocks that are looming ahead for each team member.

Object & Method Designs

Below are the object/method designs of the new improved system.

Name:	void checkPlate(string plate)
Purpose:	This function runs a license plate in the database when license plate recognition software locks onto a license plate. The plate checked to see if it is tied with a valid parking permit or if the plate is wanted by police.
Parameters:	plate = license plate of a vehicle.
Code:	Search license plate in database If license plate found Print "search successful!" Check is plate is wanted If plate wanted Save image Alert campus police Check if permit is valid If permit not valid Save image Issue citation If license plate not found Print "error: search successful"

Name:	void validatePayment(rfid student);
Purpose:	This function validates the payment of the RFID automated parking payment solution when vehicle passes through RFID sensor.
Parameters:	student = students RFID tag that the payment is being validated with.
Code:	If student has pre-paid type RFID tag Charge amount from account If account is empty Notify campus parking services Issue charges owed Else if student has pay as you go type RFID tag Charge students creditcard If credit card declined Notify campus parking services Issue charges owed

Name:	void updateOccupyStatus(int x);
Purpose:	This function updates information of a parking spot when vehicle passes or leaves parking sensor. It will be used to update the count on parking displays and exact location on the mobile application.
Parameters:	x = a specific parking spot
Code:	If vehicle goes over parking sensor (x) Set status to occupied Increment parking display counter Else if vehicle leaves parking sensor (x) Set status to available Decrement parking display counter

Name:	void getOccupyStatus(int x);
Purpose:	This function returns the occupancy status of a parking spot which will be used in the mobile application.
Parameters:	x = a specific parking spot
Code:	If status is occupied Return occupied Else if status is available Return available

Name:	void createVehicle(string sID, string sMake, string sColor, string sPlate);
Purpose:	This function inserts vehicle information into the database using SQL code.
Parameters:	sID = ID of student sMake = make of vehicle sColor = color of vehicle sPlate = license plate of vehicle
Code:	INSERT INTO VEHICLES (id, color, make, plate) VALUES (sID, sMake, sColor, sPlate);

Controls

Below are the controls for our new improved system to ensure it runs smoothly.

Control	Description
Logs	Each action will be logged in the system so problems can be easily debugged by a sysadmin
Payment Validation	To ensure reliable payment, the system will check if payment information is valid when students first set up their RFID tag with their credit card information. If there is any error inputting credit card or the credit card is rejected, the system will display an error message.
Online Forms	To ensure operations go smoothly, students and faculty will be able to report the problems they come across and be able to make suggestions to improve the new system.
Checksum data validation	To ensure data integrity in the system, data will be validated through a checksum. This is very important as data will be coming in and out of the cloud.
Security controls	To ensure data has not been modified maliciously, if data needs to be updated manually for whatever reason, the user will need permission.

Test Plan

Below is the test plan for our new improved system.

Step	Navigation	Passed	Comment	Test Description
1	Camera system	Y	N/A	Test a vehicle to see if the camera recognizes the license plate and runs it through the system. Furthermore, test if license plate recognition is working in the dark.
2	Parking Structure	Y	N/A	Test to see if the structure detects correct amount of cars in each level and displays it correctly
3	Parking Lot	Y	N/A	Test to see if each parking lot is detecting the correct amount of cars and displays it correctly.
4	Mobile Application	N	Available parking is not being updated on the mobile application.	Test to see if the mobile application is refreshing the correct available parking spots. This should be done by driving over a parking sensor repeatedly.
5	Amazon Web Services	Y	N/A	Test to see if all data is being sent to the database hosted on Amazon web services.

[Summary Report]

In short, we are trying to create a more efficient system for parking on campus. This system will give more control and information to students and staff. For students who do not wish to pay prematurely for a pass, they have a much better option for parking. While students who already benefit from a prepaid pass, they now have the ability to enjoy a mobile app that can help them find a parking space. Lastly, for campus parking police, they now have a easier way to track and process violations.

Implementing this plan will take collaboration from students and staff, but will be achievable much easier with the plan put in place to start during the summer months when most students are not in session.