Raad Khan  
System Analysis and Design  
Final Project

1. **Goal**

Create an application that allows the users to find free/cheap parking spaces near their home in NYC

1. **Objectives:**

* Build a system that can connect real-time with satellite imaging all over NYC
* Build a system that can hold information of at least 40,000 users at launch and be accessed by them using secure internet connections
* Build a system that can be accessed through a user-friendly app for iPhones and Androids that can view the map of NYC based on the satellite imaging

1. **Scope Statement:**

* Create a platform that can store user information such as name, address, car model, car dimensions, license plate number and is user friendly
* The system shall be connected to the internet
* The system shall be able to collect satellite imaging of NYC clearly of the streets
* The system shall be able to find open parking areas in the streets
* System should be able to map out parts of the city with free and paid parking based on the time of day
* System shall be able to refresh and present updates real-time
* System should store information of every major make and model of cars and their dimensions
* System should alert users if they need to move parked car
* System shall allow users to create profiles
* System shall allow users to select monthly or yearly payments’
* The system should be able to create a secure way to accept and process payments using credit/debit cards
* The system shall allow users to access and add information to a forum where they can share ideas
* The system should have a search function in the forum to look for existing forums based on keywords they search for
* The system should allow administrative privileges to admins and let them delete accounts and comments in the forum

1. **Out of scope:**

* The system will not work outside of NYC
* The system cannot find parking based on any location, but only near the user’s address
* The system will not have a pay-by-each-use payment method
* The system will not be able to store multiple cars for each user account on the app

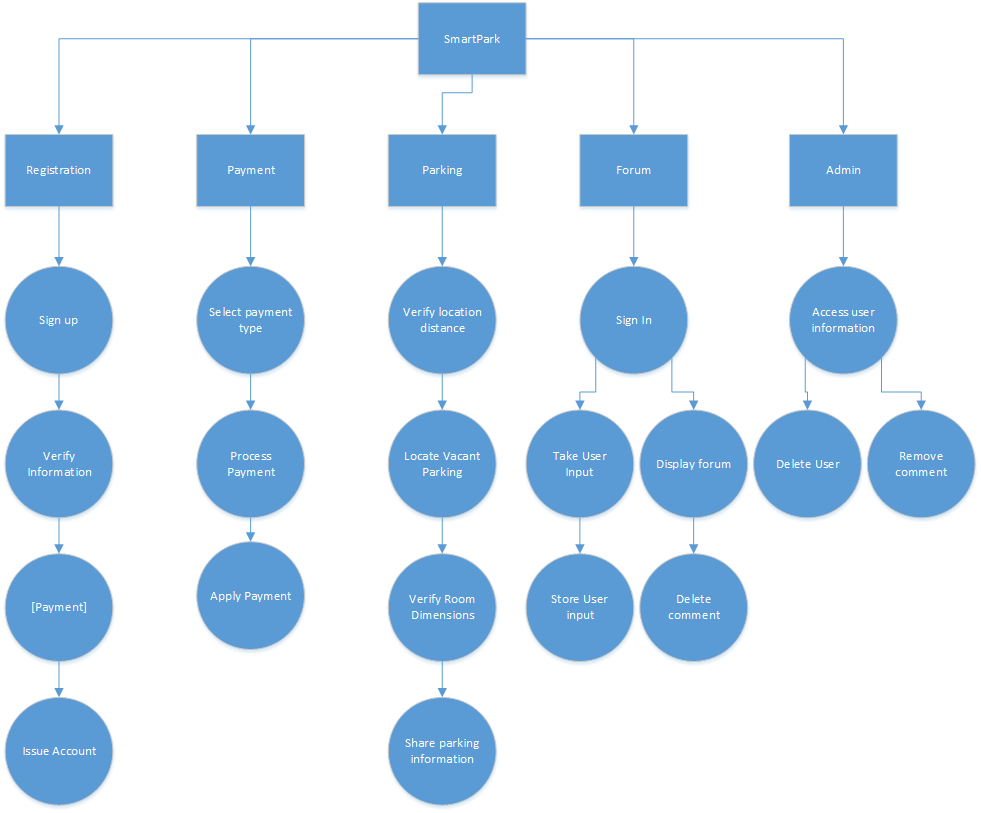
1. **Risk:**

* The technology to create an application like this is not industry tested, thus creating something streamlined may push the scope, increase costs during production and push the timeline further

1. **Constraints:**

* The budget is set to $400,000 set by the owner, the budget cannot be exceeded
* The timeline for this system is 6 months. This will prevent a competitor from beating us to market
* The scope has been reduced to necessary items and cannot be reduced further

1. **Decomposition Diagram:**



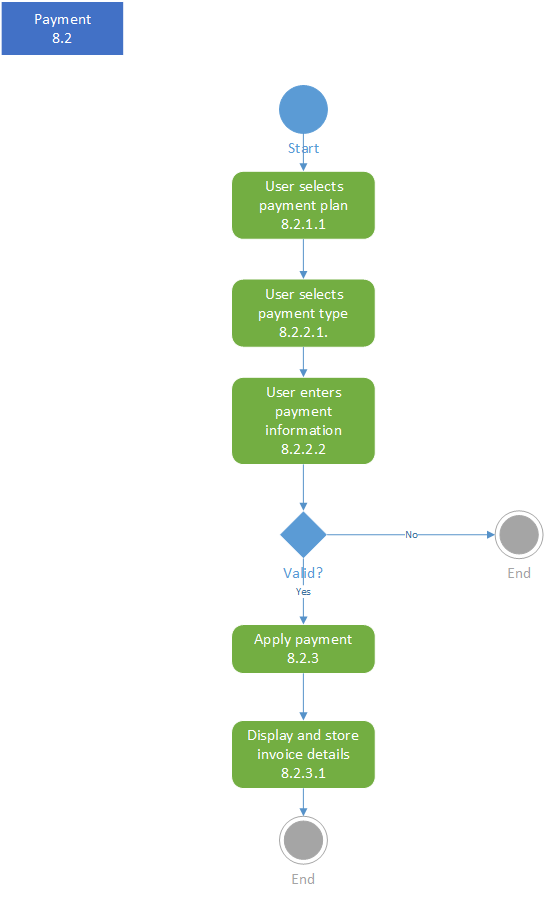
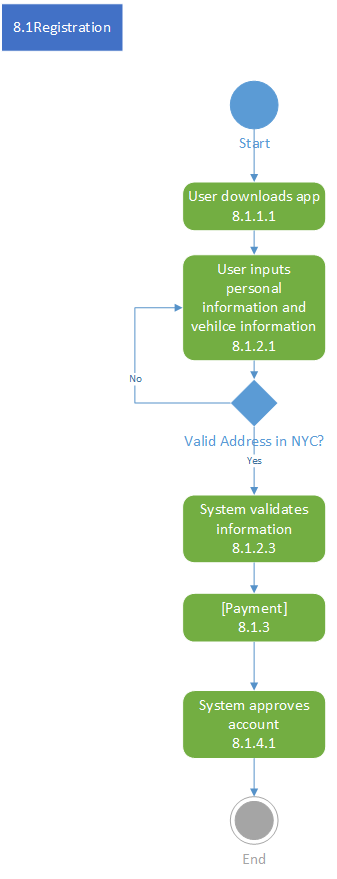
1. **Functional Requirements:**
   1. Registration
      1. Sign Up
         1. User is to decide to register for an account by downloading the app
      2. Verify Information
         1. User is to enter their full name, address, car model, car make, license plate number, phone number and email
         2. The dimensions of the car should be verified by the user off the system database for information on each car
         3. The address should be verified to ensure it is a location in NYC and within the range assigned by the system
         4. Phone number should be verified by the user, system will validate a unique phone number is used for the account
      3. [Payment]
      4. Issue Account
         1. Once all the information is verified, an account should be issued with a username and password selected by the user, using which the user can access the app at any point with an internet connection
   2. Payment
      1. Select payment type
         1. User will get to select if they want a monthly payment plan or a yearly payment plan
      2. Process payment
         1. The system should allow the user to enter pick their desired payment method
         2. The system should allow the user to enter their payment details
      3. Apply payment
         1. The system should confirm payment has been processed and transaction can be verified for each purchase
         2. System should store record of each transaction
   3. Parking
      1. Verify location distance
         1. Based on the address of the user, the system will verify if there is parking available nearby up-to 5 blocks away
         2. The user should be allowed to select if they are looking for free or paid parking
      2. Locate vacant parking
         1. Based on satellite imaging, the system will search for an empty parking spot on the streets within a radius of 5 blocks from the address
         2. If no open locations are found, the user will be given the options to keep refreshing the search every 5 minutes until a parking location is found
      3. Verify room dimensions
         1. Once a parking spot is found, the system will verify that the parking spot has dimensions that will safely support the dimensions of the vehicle used by the user
      4. Share parking information
         1. Once all the requirements are met, the system will present the user with the location of the parking spot
         2. The system will present information on the parking spot itself, if it is paid or free, and how long they are allowed to park there for
   4. Forum
      1. Sign in
         1. User can sign in to the app with their username and password and access the forum
      2. Take user input
         1. The user is allowed to start a thread in the forum or comment on an existing thread
      3. Store user input
         1. The system should be allowed to store the information onto the forum for later viewing by admin or other users
      4. Display forum
         1. The user can choose to access the different threads and comments available on the forum, based on their specific questions
         2. The user should be able to use a search function to find threads with keywords for what information they need
      5. Delete comment
         1. The user can access their previous comments or threads and are allowed to delete them
   5. Admin
      1. Access user information
         1. Admins should have access to users on the app
         2. The system should have a search function to lookup user by username, email, phone number, name, car make and model or license plate number
      2. Delete User
         1. The admin should have the privilege to delete a user’s account
      3. Delete Comment
         1. The admin can access the different forum threads and comments and delete given threads or comments
2. **Nonfunctional requirements**
   1. Security
      1. The system shall have SSL
      2. The system shall encrypt all personal data
      3. The system shall have a secure way to perform payments and transaction
   2. Storage Space
      1. The system shall have storage space for 40,000 user account information
      2. The system shall have storage space for user payment methods
      3. The system shall have storage for information on dimensions of each unique car make and model
   3. Appearance
      1. The system should have user-friendly and easy to navigate app
   4. Backup/recovery

9.4.1. The system shall have a live/hot backup

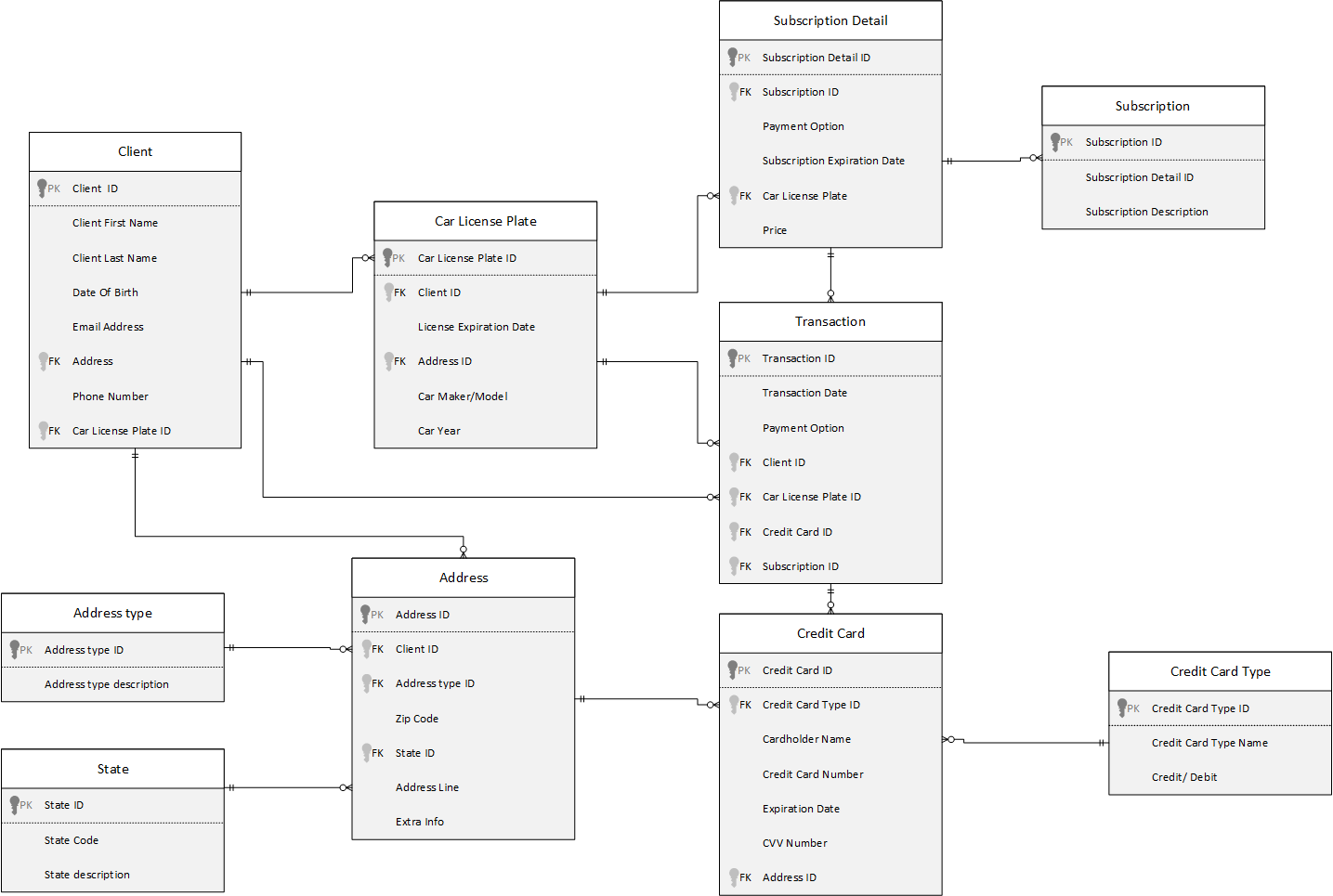
**10. Stakeholders:**

* Jack
* Investors
* Employees
* Directors
* Clients

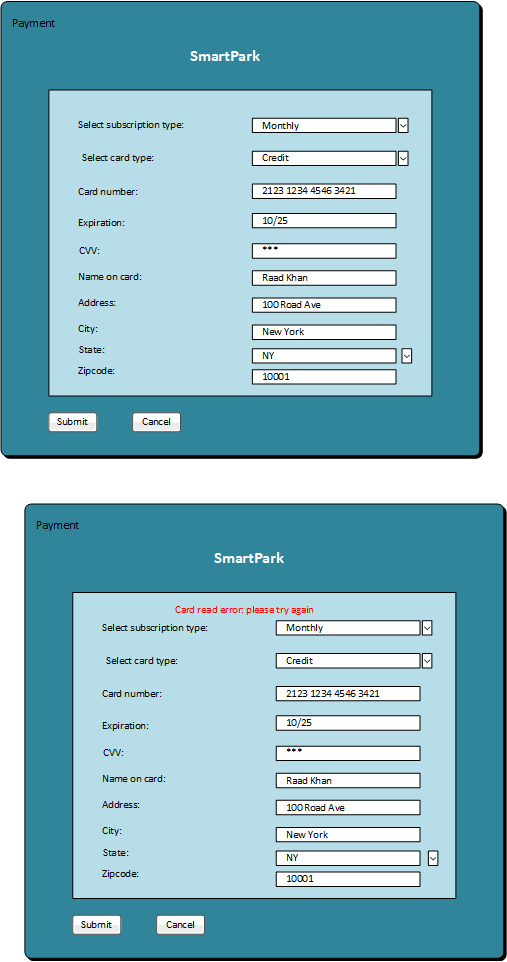
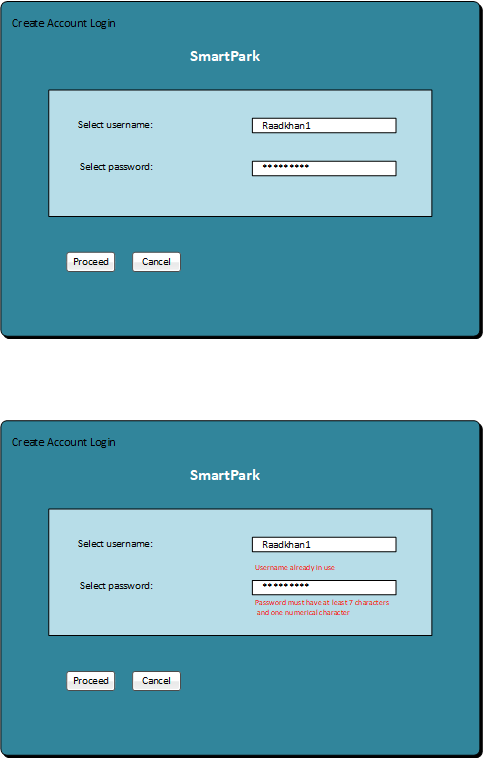
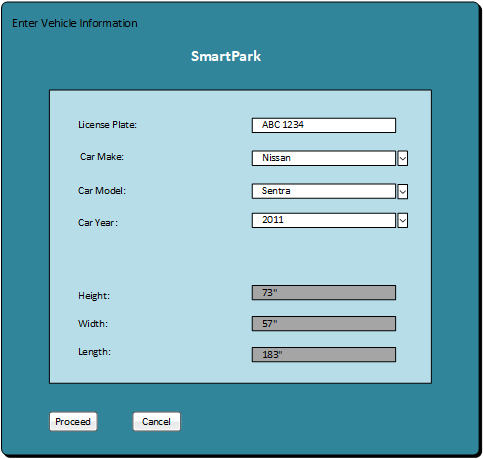
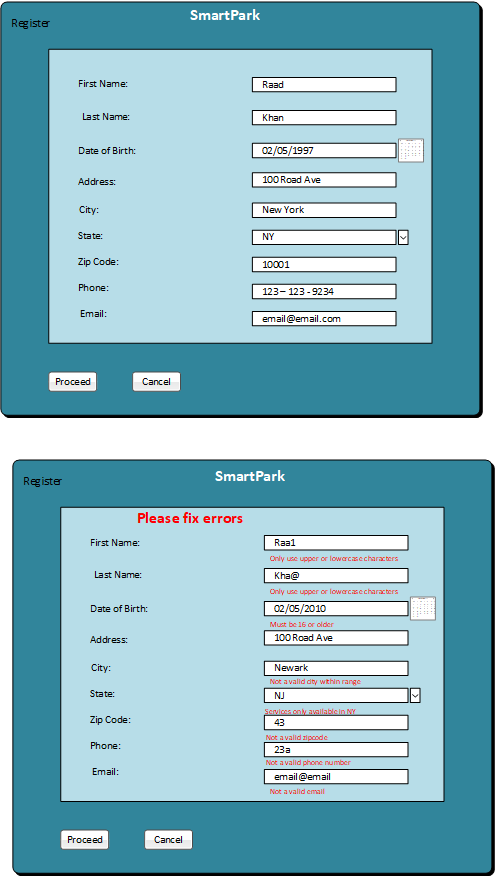
1. **Activity Diagram:**



1. **Data Model**



1. **Wireframe:**



1. **Use Case Statements**

|  |  |  |
| --- | --- | --- |
| Line Number | User Action | System Action |
| 1 | User sets up app | System displays Register page |
| 2 | User Enters First Name | --- |
| 3 | User enters Last Name | --- |
| 4 | User enters Date of Birth | --- |
| 5 | User enters Address | --- |
| 6 | User enters City | --- |
| 7 | User selects State | --- |
| 8 | User enters Zipcode | --- |
| 9 | User enters Phone | --- |
| 10 | User enters Email | --- |
| 11 | User clicks proceed | System validates information System stores record |
| 12 |  | Control goes to Enter Vehicle Information System displays Enter Vehicle Information |
| 13 | User enters License Plate | --- |
| 14 | User selects Car Make | --- |
| 15 | User selects Car Model | --- |
| 16 | User selects Car Year | --- |
| 17 |  | System displays Height, Width, Length of vehicle |
| 18 | User clicks proceed | System validates information System stores record |
| 19 |  | Control goes to Create Account Login System displays Create Account Login |
| 20 | User enters Username | --- |
| 21 | User enters Password | --- |
| 22 | User clicks proceed | System validates information System stores record |
| 23 |  | Control goes to Payment System displays Payment |
| 24 | User selects Subscription type | --- |
| 25 | User selects Card Type | --- |
| 26 | User enters Card Number | --- |
| 27 | User enters Expiration | --- |
| 28 | User enters CVV | --- |
| 29 | User enters Name on Card | --- |
| 30 | User enters address | --- |
| 31 | User enters city | --- |
| 32 | User selects State | --- |
| 33 | User enters Zipcode | --- |
| 34 | User clicks Submit | System validates information System stores record |
| 35 |  | System processes payment |
| 36 |  | System creates account |
| 37 | End Use Case |  |
| 11.1 |  | System displays error for incorrect information |
| 11.2 | User corrects errors | --- |
| 11.3 |  | Control returns to line 11 |
| 18.1 |  | System displays error for incorrect License plate |
| 18.2 | User corrects errors | --- |
| 18.3 |  | Control returns to line 18 |
| 22.1 |  | System displays error for invalid username or password |
| 22.2 | User corrects errors | --- |
| 22.3 |  | Control returns to line 22 |
| 34.1 |  | System displays error for incorrect card information |
| 34.2 | User corrects errors | --- |
| 34.3 |  | Control returns to line 22 |