**Team 2 - ParkEZ (Fall 2023)**

**Use Cases I3**

**Team 2 - ParkEz (Fall 2023)**

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision #** | **Created By** | **Revision Notes** |
| 11/06/2023 | 1.0 | Krunal Bhavsar | Initial Draft |
| 11/06/2023 | 1.1 | Tom Alex | 5.1 Define Payment Method |
| 11/06/2023 | 1.2 | Tom Cookson | 3.3 Access Occupancy Remotely |
| 11/06/2023 | 1.3 | Shubham Mittal | 4.2 Modify Ad |
| 11/06/2023 | 1.4 | Srija Reddy Sripathi | 3.1 Search and select Slot |
| 11/06/2023 | 1.5 | Chetan Basnet | 3.2 View Occupancy |
| 11/06/2023 | 1.6 | Suruchi Patil | 4.3 View Ads Statistics |
| 11/07/2023 | 1.7 | Ruchi Sharma | 2.1 Track Occupancy |
| 11/07/2023 | 1.8 | Swayambhu Dhuri | 5.2 Validate Payment |
| 11/18/2023 | 1.9 | Tom Cookson | Post-critique review |

**Iteration 3**

# **UC.03.01   Search and Select Lot**

|  |  |
| --- | --- |
| **Actor(s):** | Car Parking User, Lot Business User, Advertising User, Lot Business Manager, Advertising Manager, Customer Support Manager, Accounting Manager |
| **Short Description:** | This Use case allows the Car parking User for Searching and Selecting a parking lot near the destination specified by the user. |
| **Pre-conditions:** | 1. There should be available parking lots in the system. 2. The information about available lots should be up-to-date and accurate. 3. The Actor has access to the internet and the ParkEz platform to perform the search. |
| **Post-conditions:** | 1. The Actor clicked or has not clicked (they didn’t find the lot they needed) the lot they wanted to patronize. 2. Once the Actor selects the lot, they are navigated to further details of the parking lot showing occupancy of parking spaces. |
| **Frequency of Use:** | High |
| **Normal Flow­­ of Events:** | |
| 1. This use case begins when Actor inputs a search phrase (e.g., "NY") into the search field to find parking lots nearby. [JP1: CS, CA] 2. System displays a list of available parking lots sorted by distance to the user, each with the address. [JP2: DF-In, CL, PF] 3. Actor reviews the list of available parking­ lots. 4. Actor selects a parking lot from the list and the use case ends. | |
| **Alternative Flows:** | |
| A1: “No Search Results”  If No lots found matching criteria, perform A1, from step 3.   1. System returns no results based on the Actor’s specified criteria after searching for available lots 2. System provides a message informing the user that no lots match their criteria. 3. Actor is given the option to modify their search criteria and the use case continues from step 1. | |
| **Exceptions:** | |
| E1: “System Issues”  If System issues or unavailable, perform E1, from step 2.   1. System detects the unavailability or technical issues during the user's attempt to access the lot search feature. 2. Actor is advised by the error message to try again later. 3. System provides timeframe for when system is expected to be available again may be provided. 4. System offers alternative means of contacting support for further assistance if possible and the use case ends. | |
| **<<Include>> Relationships:** | None |
| **<< Extend>> Relationships:** | None |
| **Assumptions:** | The parking lot can patronize, and parking spaces are open to use. (e.g., No construction taking place) |

**UC.03.02 View Occupancy**

|  |  |
| --- | --- |
| **Actor(s):** | Car parking user, Lot Business User, Advertising User, Lot Business Manager, Advertising Manager, Customer Support Manager, Accounting Manager |
| **Short Description:** | The use case allows car parking user to select the best parking spot from a selected parking lot. |
| **Pre-conditions:** | The user must select a parking lot out of the lot available in the ParkEz application. |
| **Post-conditions:** | The user will be able to see the best available parking spot if available. |
| **Frequency of Use:** | High |
| **Normal Flow of Events:** | |
| 1. This use case begins when Actor selects desired parking lot. [JP1: EN-In, EN-Et] 2. Actor sees available and used parking spots. [JP2: CS, SI-In, PF] 3. System ensures camera feed is periodically updating and there is no unexpected gap in data. [JP3: CN, DF-In] 4. System caches data on browser. [JP4: CA] 5. System uses algorithm to select the best parking spot out of the available spots and the use case ends. [JP5: ER] | |
| **Alternative Flows:** | |
| A1: “Spot Not Found”  If Actor cannot find spot, perform A1, from step 2.   1. Actor select selects another parking lot and the use case continues from step 1. | |
| **Exceptions:** | |
| E1: “Camera Unavailable”  If Camera is down or obscured, perform E1, from step 2.   1. Actor not able to select any parking spot. 2. System gives the actor a message notifying of difficulties and the use case ends. | |
| **<<Include>> Relationships:** | None |
| **<< Extend>> Relationships:** | None |
| **Assumptions:** | The parking spaces that are considered are the one that is within the range of the camera. There maybe parking spot outside the range of camera that ParkEZ will not take into consideration while calculating best parking spot |

**UC.03.03 Access Occupancy Remotely**

|  |  |
| --- | --- |
| **Actor(s):** | Car Parking User, Lot Business User, Advertising User, Lot Business Manager, Advertising Manager, Customer Support Manager, Accounting Manager |
| **Short Description:** | This use case allows Car Parking Users to access recent public parking lot occupancy information and webcam images, which can be embedded on a third-party business's website. |
| **Pre-conditions:** | The business owner must have a parking lot that is monitored by ParkEz's system.  The business owner must have an editable website (HTML, etc.) |
| **Post-conditions:** | The Car Parking User is able to see the current parking lot occupancy.  The webcam images with overlays showing occupied, vacant, and the recommended best spot are viewable on the business website.  The parking lot's occupancy data and images are updated at the expected interval on the business's website. |
| **Frequency of Use:** | High (as users regularly check for parking availability and recommendations before arriving at the location). |
| **Normal Flow of Events:** | |
| 1. This use case begins when Actor navigates to a business website with embedded ParkEz parking information. [JP 1: CS] 2. Actor views the current occupancy information, which includes the number of occupied and vacant spots. [JP 2: DF-In, SI-In, CN] 3. Actor views the webcam images that show the actual parking lot with indicators highlighting occupied and vacant spots. [JP 3: DF-Out, CA, PF] 4. System also marks the recommended best spot based on current occupancy and distance to the business entrance. [JP 4: CL, ER] 5. Actor makes a parking decision based on the information provided and the use case ends. | |
| **Alternative Flows:** | |
| A1: “Full Parking Lot”  If Parking lot is full, perform A1, from step 2.   1. System will not provide a best spot recommendation. 2. System recommendation is: “None” and the use case ends. | |
| **Exceptions:** | |
| E1: “Image Transmission Down”  If Webcam or image transmitter down, perform E1, from step 2.   1. System displays the last available image and parking lot information: 2. Actor sees the time of the information which will inform the user it is not recent and the use case ends. | |
| **<<Include>> Relationships:** | None |
| **<< Extend>> Relationships:** | None |
| **Assumptions:** | The computer vision system and machine learning algorithms are accurate in determining parking spot occupancy.  There is the correct, expected delay between the actual occupancy changes and the update on the business's website.  The business's website can handle the integration without significant performance degradation. |

**Iteration 1**

# **UC.05.01 Define Payment Method**

|  |  |
| --- | --- |
| **Actor(s):** | Advertising User, Lot Business User |
| **Short Description:** | This use case begins when the Actor decides to define or update a payment method and ends when the payment method has been successfully added, updated, or removed. |
| **Pre-conditions:** | Actor should be logged into the system. |
| **Post-conditions:** | Actor’s payment methods list is updated.  The default payment method is set if chosen.  System records the changes for audit and future transactions. |
| **Frequency of Use:** | Average |
| **Normal Flow of Events:** | |
| 1. This use case begins when the Actor navigates to 'Management Payment Methods' after clicking 'Billing'. [JP 1: EN-In, EN-Et] 2. System displays the 'Management Payment Methods' screen. [JP 2: CA] 3. Actor adds a new credit card with details and sets it as a payment method. [JP 3: CS] 4. System verifies the card details with Stripe and updates the payment methods list. [JP 4: EN-In] 5. Actor optionally sets one of the payment methods as the default. [JP 5: DDD] 6. System updates the default payment method accordingly. [JP 6: CC] 7. Actor removes any unwanted payment methods. 8. System confirms the removal and updates the list both in the system and Stripe. [JP 7: SI-In] 9. Actor logs out or navigates away from the 'Management Payment Methods' screen and the use case ends. [JP 8: CA] | |
| **Alternative Flows:** | |
| A1: “Invalid Payment Method”  If Invalid payment method entered, perform A1, from step 3.  1. System recognizes the card as a duplicate or invalid.  2. System displays an appropriate error message, prompts the user to enter a different card and and the use case continues from step 3. | |
| **Exceptions:** | |
| E1: “Failed to Connect”  If Network issues or Stripe API failure, perform E1, from step 2.  1. System fails to connect to Stripe or experiences a timeout.  2. System displays a network error message, asks the Actor to try again later and the use case ends.  E2: “Actor Failed to Complete”  If Actor navigates away or logs out before completing, perform E2, from step 3.  1. Actor closes the browser or logs out before saving changes.  2. System does not update unsaved changes in Stripe, and the use case ends. | |
| **<<Include>> Relationships:** | None |
| **<< Extend>> Relationships:** | None |
| **Assumptions:** | All interaction with the Stripe API follows the secure protocols and handles sensitive data according to PCI DSS standards. |

**UC.05.02 Validate Payment**

|  |  |
| --- | --- |
| **Actor(s):** | Lot Business User, Advertising User |
| **Short Description:** | This use case begins when the system needs to validate a payment and ends when the payment validation is completed. |
| **Pre-conditions:** | The system has initiated a payment validation process. |
| **Post-conditions:** | The payment validation result is recorded.  The system may proceed with the transaction if the payment is validated successfully. |
| **Frequency of Use:** | Average |
| **Normal Flow of Events:** | |
| 1. The use case begins when Actor triggers the payment validation process for a specific transaction. 2. System communicates with the Stripe API to initiate the payment validation. [JP 1: CA, CS] 3. System uses Stripe to process the payment validation request and checks the validity of the payment method provided. 4. System receives validation result from Stripe. [JP 2: SI-In] 5. System records the payment validation result. 6. System validates the payment successfully and transaction is logged. 7. System completes payment validation process, and the use case ends. | |
| **Alternative Flows:** | |
| A1: “Payment validation timeout”  If Payment validation timeout, perform A1, from step 4.   1. Stripe does not respond within a reasonable time. 2. System displays a timeout error message. 3. System may retry the payment validation and the use case continues from step 1.   A2: “Payment not successfully validated”  If Payment not successfully validated, perform A2, from Step 6.   1. System informs the relevant actors about the failed validation. 2. System does not complete transaction. 3. System updates the payment status as "invalid”, notifies the Actor and the use case ends. | |
| **Exceptions:** | |
| E1: “Stripe API is unavailable”  If Stripe API is unavailable, perform E1, from step 2.   1. System fails to connect to the Stripe API. 2. System displays an error message, asks the user to try again later and the use case ends. | |
| **<<Include>> Relationships:** | None |
| **<< Extend>> Relationships:** | None |
| **Assumptions:** | All interactions with the Stripe API follow secure protocols and handle sensitive data according to PCI DSS standards. |

**Iteration 2**

# **UC.02.01 Track Occupancy**

|  |  |
| --- | --- |
| **Actor(s):** | Lot Business User, Lot Business Manager, Customer Support Manager |
| **Short Description:** | This use case begins when a Lot Business User or Manager logs into the system to monitor and track the occupancy of parking spaces. It involves the collection and processing of data from external sources to provide real-time occupancy information and ends when the system successfully updates and displays this information. |
| **Preconditions:** | Parking lot management system is operational.Cameras or sensors are installed and functional.System has been calibrated with the layout of the parking lot. |
| **Postconditions:** | Occupancy data is accurately reflected in the system.System’s occupancy information is current and accurate.Advertisements are correctly associated with their respective lots. |
| **Frequency of Use:** | High |
| **Normal Flow of Events:** | |
| The use case begins when Actor logs into the system and navigates to the ‘Track Occupancy’ section. [JP 1: SI-In, CS, CA]System displays the current occupancy status using data from cameras or sensors. [JP 2: ER, CN, SI-In, PF]System processes the incoming data and updates the occupancy information using the optimal spot formula. [JP 3: CL]Actor reviews the updated occupancy data for accuracy. [JP 4: CL]Actor uses the occupancy data to determine lot fullness and make operational decisions. [JP 5: CL]System confirms the accuracy of data and displays it on the GUI.System logs the transaction for future audit and reference. [JP 7: DF-Out]Actor logs out or navigates away from the 'Track Occupancy' section and the use case ends. | |
| **Alternative Flows:** | |
| A1: “Occupancy data is invalid”If Occupancy data is invalid, perform A1, from step 2.System recognizes the data as invalid or inconsistent with the expected format or values. [JP 8: CL]System displays an appropriate error message indicating the invalid occupancy data, prompts the user to check the sensors or cameras and the use case ends. | |
| **Exceptions:** | |
| E1: “Network or sensor API failure”If Network or sensor API failure, perform E1, from step 2.System fails to receive data from cameras or sensors due to network issues. [JP 10: SI-In]System displays a network error message and asks the Actor to verify the hardware status or try again later, and the use case ends.E2: “Actor exits prematurely”If Actor exits prematurely, perform E2, from step 4.Actor navigates away or logs out before the system saves changes.System does not record unsaved data, retains the previous occupancy data­­­­­ [JP 13: SI-In] and the use case ends. | |
| **<<Include>> Relationships:** | None |
| **<< Extend>> Relationships:** | None |
| **Assumptions:** | All camera and sensor data are reliable and transmitted securely.The system's calculations for occupancy are precise and updated in real-time.Ads are correctly configured to be associated with specific lots and their occupancy status. |

# 

# **UC.04.02 Modify Ad**

|  |  |
| --- | --- |
| **Actor(s):** | Advertising User, Advertising Manager, Customer Support Manager |
| **Short Description:** | This use case allows the Actors to be able to make modifications in their account’s associated ads such as Name, Start Date, End Date, URL, Top Banner Image 1-3 and Image Change Interval and check 1 or more different associated lots (where the ad will show) and save them by clicking submit. |
| **Pre-conditions:** | Actor must have an account.  Actor must have pre-filled details to modify. |
| **Post-conditions:** | The actor must be able to see the account details. Actors must be able to edit information or update the details. (Actors can update the following information: Name, Start Date, End Date, URL, Top Banner Image 1-3 and Image Change Interval and check 1 or more different associated lots (where the ad will show)) |
| **Frequency of Use:** | Average |
| **Normal Flow of Events:** | |
| 1. This use case begins when Actor logs in the account. [JP 1: EN-In] 2. Actor clicks on the edit button on the dashboard to make modifications to the details. [JP 2: EN-Et] 3. Actor sees all the information edited or saved previously. [JP 3: DF-In, CA, PF] 4. Actor can update the information: Name, Start Date, End Date, URL, Top Banner Image 1-3 and Image Change Interval and check 1 or more different associated lots (where the ad will show). [JP 4: PF] 5. Actor clicks submit to save the information. [JP 5: CA, DF-Out] 6. System validates the information. [JP 6: CC] 7. System displays a confirmation message. 8. Actor confirms all data fields and images match what was entered and the use case ends. | |
| **Alternative Flows:** | |
| A1: “Not able to update ad”  If Actors not able to update ad, perform A1, from step 5.   1. Actors are invited to contact customer care to get help with their account and the use case ends.   A2: “Invalid data”  If Actor submits invalid data, perform A2, from step 5.   1. System detects invalid input or incomplete required fields. 2. System displays an error message detailing the issue and the use case continues from step 3.   A3: “No data date”  If Actor submits no data date, perform A3, from step 5.   1. System will show errors, display fill in messages and the use case continues from step 3. | |
| **Exceptions:** | |
| E1: “Servers are down”  If Servers are down, perform E1, from step 3.   1. Actor receives directions to contact the customer care for urgent modifications or wait for servers to be up, and the use case ends. | |
| **<<Include>> Relationships:** | None |
| **<< Extend>> Relationships:** | None |
| **Assumptions:** | 1. The advertiser has the proper authorization to edit advertisements. 2. The system is capable of real-time updates of advertisements. 3. All associated lots are part of the system's managed properties. |

**UC.04.03 View Ad Statistics**

|  |  |
| --- | --- |
| **Actor(s):** | Advertising User, Customer Support Manager, Advertising Manager |
| **Short Description:** | This use case allows the advertisers to see the statistics of the ads. |
| **Pre-conditions:** | 1. The user must be logged in to the system and have the appropriate permissions to view ad statistics. 2. The ad for which the user wants to view statistics must exist in the system. 3. The ad must have been published. 4. The ad must have at least one statistic to view. |
| **Post-conditions:** | 1. The user should be able to see all of the statistics for the specified ad, including impressions, clicks, click-through rate (CTR), and conversions. 2. The statistics should be accurate and up-to-date. 3. The user should be able to easily understand the statistics and use them to make informed decisions about their ad campaigns. |
| **Frequency of Use:** | Average |
| **Normal Flow of Events:** | |
| 1. This use case begins when Actor logs into the ad statistics system. 2. System performs an entitlement check to verify the actor's permissions. [JP1: EN-In, EN-Et] 3. System monitors performance of the login process for efficiency. [JP2: PF] 4. Actor selects the ad for which they want to view statistics. [JP3: DF-In] 5. System accesses the cache to retrieve any recently viewed ad statistics. [JP4: CA] 6. System retrieves the statistics for the selected ad. 7. The system's performance in fetching ad statistics is evaluated to ensure timely data retrieval. [JP5: PF] 8. System displays the statistics to the user. [JP6: DF-In] 9. Actor reviews the statistics and make any necessary decisions about their ad campaign and the use case ends. | |
| **Alternative Flows:** | |
| A1: “System displays incorrect statistics”  If System displays incorrect statistics, perform A1, from step 8.   1. Actor contacts the ad platform's support team and the use case ends. | |
| **Exceptions** | |
| E1: “Servers are down”  If Servers are down, perform E1, from step 5.  1. Actor will not be able to access data from system connect to the ad statistics system.  2. System displays error message and the use case ends. | |
| **<<Include>> Relationships:** | None |
| **<< Extend>> Relationships:** | None |
| **Assumptions:** | 1. The ad statistics system is up and running. 2. Advertiser has a valid account with the ad statistics system. 3. Actor is logged into the ad statistics system. 4. Actor has permission to view the statistics for the ad they want to view. 5. Ad has been published and served at least once. |