**Attempts to Authenticate**

**Primary Actors:** Staff, Faculty

**Stakeholders and Interests:**

* *Requestor(Staff or Faculty):* wants to authenticate into the system in order to obtain a parking permit for a parking slot. He or she would prefer to use his existing Regis login.

**Preconditions:**

* Authentication system is configured by the application

**Success Guarantee (Postconditions):**

* User is authenticated and has authorization to request parking spots.
* User sees a screen that allows him to see any existing reservation with the ability to create a new slot reservation.
* User has the ability to log off
* User session is set

**Main Success Scenario**

1. User goes to main URL where he is presented with a login screen consisting of a username and password field, along with a submit button.
2. User enters invalid credentials [Alt 1: Invalid Credentials]
3. User leaves a field blank and submits form [Alt2: Missing Required Fields]
4. Unauthorized user tries to login [Alt 3: Unauthorized User Tries to Login]
5. User enters in valid credentials and submits the form
6. User information is checked against the user table
7. System logs the user logon and sets user session
8. User is brought to the main screen for authenticated users which consists of a list of existing reservations, the ability to create a reservation, and the ability to log off. [Use Case Ends]
9. User click link to log out
10. User is brought back to login form
11. User’s session information is cleared

**Alternative Flows**

*Alt 1:* Invalid Credentials

1. Flow resumes at Step 1 with a message indicating that either the username or password is incorrect.
2. System logs the invalid login attempt

*Alt 2:* Missing Required Fields

1. Flow resumes at Step 1 with a message indicating that all fields must be filled out.

*Alt 3:* Unauthorized User Tries to Log In

1. Flow resumes to Step 1 with a message indicating that the user does not have sufficient privileges to access the system
2. System logs the invalid login attempt

**Exceptions**

If the system cannot reach the authentication system or the authentication system throws an exception, the exception is logged and the user is notified that there’s an issue with the system. The system administrator should be notified.

**Special Requirements**

Authentication form should have reasonable sized input fields and labels.

**Open Issues**

* Do we have access to the authentication system?
* If so, do we need configuration information to interact with the authentication system?
* Is there a maximum number of login attempts to avoid hacking?
* Should a captcha be used for security purposes?

**Submits a Request for a Parking Spot**

**Primary Actors:** Staff, Faculty, Administrator

**Stakeholders and Interests:**

* *Requestor (Staff or Faculty):* wants to obtain a parking permit for a specific event
* *Administrator:* wants only staff and faculty to obtaining parking permit, wants to be notified whenever a reservation is made, wants a record of who is submitting a request for a permit, wants to know which spots are taken up. The administrator can also perform the same roles as the requestor and make/update/cancel reservations

**Preconditions:**

* User is authenticated by system

**Success Guarantee (Postconditions):**

* User receives a confirmation that his request is being reviewed.
* Administrator receives an email mentioning that a request needs reviewing
* A record has been inserted into the database with request information

**Main Success Scenario**

1. User clicks the create request button
2. User is presented a form with fields appropriate for acquiring a parking reservation including a dropdown with the events to chose from.
3. There are no slots left [*Alt 1:* No Slots Left]
4. User fills out form, which includes the number of slots they’d like to request and the lot they want, and hits the submit button
5. Some required fields are not filled in [*Alt 2:* Required Fields Need Filling Out]
6. Form is successfully submitted
7. The slots may fill up between the actions of the user making his decision and hitting submit. [*Alt 1:* No Slots Left]
8. User receives a confirmation that his request is being reviewed
9. Administrator(s) and requestor receives an email that a request needs reviewing [*Use Case Ends*]
10. User clicks the cancel button next to his or her reservation
11. A popup shows up asking the user if he is sure he wants to cancel the reservation
12. User clicks OK in the popup
13. Screen is updated with parking reservation removed
14. Information is about the reservation is cleared from the database
15. First user on waiting list is emailed to let him know that a slot has opened up
16. The administrators receive an email notifying them of the cancellation.

**Alternative Flows**

*Alt 1:* No slots left

1. User is notified that there are no available slots and to check back later to see if a slot has opened up.
2. User is given an option to be put on a waiting list
3. If user, choose to be put on waiting list, but user into a waiting list table.

*Alt 2:* Required Fields Needs Filling Out

1. Flow resumes at step 1 with an error message next to each empty required field and existing filled out fields pre-populated

**Exceptions**

If the system is not able to save the reservation, log the error and send an email to the system administrator, and let the user know that there is a system issue. Use case ends.

**Special Requirements**

Form should have reasonable sized input fields and labels. Required fields should be clearly marked with asterisks on the labels.

**Open Issues**

* None

**Parking Requested Reviewed**

**Primary Actors:** Administrator, Requestor (Staff or Faculty)

**Stakeholders and Interests:**

* *Administrator:* Reviews information about a parking requests and approves or declines it.
* *Requestor:*Submitted a request for a parking slot and is looking to obtain a permit

**Preconditions:**

* Administrator has logged in with an administrator account
* Staff or faculty has submitted a request for a parking slot

**Success Guarantee (Postconditions):**

* Staff or faculty given a confirmation screen that the user has been granted or denied a permit
* Permit information, such as a unique number, has been generated and put in the database
* Staff or faculty receives an email with an attached permit

**Main Success Scenario**

1. Administrator receives an email saying that someone has requested a parking permit
2. Administrator logs into portal using his or her Regis credentials
3. Administrator sees a list of open requests and clicks one
4. Administrator sees information about the request taken from the form the requestor filled out
5. Administrator hits decline button [*Alt 1:* Administrator Declines Permit]
6. Administrator reviews information, enters the gate code for the permit into a textbox, and hits an approve button
7. Administrator receives a popup asking if he’s sure he wants to approve the request
8. Administrator hits cancel on the popup [*Alt2:* Administrator Cancels Approval]
9. Administrator hits OK on the popup
10. Administrator receives a confirmation screen saying that the user has been approves for parking slot x with and a permit number
11. Approval is stored in the database
12. An email is sent out by the system to the requestor and administrator with parking permit information and gate code. [*End Use Case*]

**Alternative Flows**

*Alt 1:* Admin Declines Permit

1. Administrator receives a popup asking if he’s sure he wants to decline the permit
2. Administrator hits cancel on the popup. [*Alt 3:* Administrator Cancels Decline]
3. Administrator hits OK on the popup
4. Administrator receives a confirmation screen saying that the user has been declined for a parking slot
5. An email is sent out by the system to the requests with a message saying the parking request has been declined [*End Use Case*]

*Alt 2:* Administrator Cancels Approval

1. Flow resumes at step 4

*Alt 3:* Administrator Cancels Decline

1. Flow resumes at step 4

**Exceptions**

* If the system is not able to save the status of the request (accepted/declined), log the error and send an email to the system administrator, and let the administrator know that there is a system issue. Use case ends.
* If the email system is offline, log the error into the database and let the administrator know so that he can resubmit the email or send it manually

**Special Requirements**

* None

**Open Issues**

* Should the permit be sent as an attachment or as text in the email?
* Should the permit even be sent in the email or should a link be sent to the requestor instructing him to log into his or her account to obtain the permit (more secure but more steps to implement)
* Does the administrator choose a parking slot from a list of available slots or just enter a slot number into a freeform text field.