=================

Use case name:

Login

------------------------

Actors:

Disabled Driver

------------------------

Flow of events:

1. Present form that collects user input for username and password and provides options for "forgot username", "forgot password", "register new user" and "login"
2. If forgot username, present form to collect email or phone from user, send matching username to user, redisplay the login form
3. If forgot password, present form to collect email or phone from user, generate new random password and save to user record, send new password to user, redisplay the login form
4. If login, validate username/password combination; if match, allow access and display home form, else display "fail" message, redisplay the login form
5. If register, display "register new driver" form

------------------------

Entry Conditions:

User is registered with system

------------------------

Exit Conditions

User is logged in

------------------------

Quality Requirements:

Valid username/password provided

=================

Use case name:

Register Driver

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user/driver is presented with a form requesting: preferred username, full name, student/faculty id, vehicle info (license plate, make, model, year) and preferred password
2. The user submits the form
3. The system validates the data and checks for a duplicate username
4. If duplicate username or invalid data items, present warning and repeat registration attempt
5. Else, allow access, update driver profile and display the home form

------------------------

Entry Conditions:

User is NOT registered with system

------------------------

Exit Conditions

User is registered

------------------------

Quality Requirements:

None

=================

=================

Use case name:

Update Driver Profile

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user/driver is presented with a form requesting: full name, student/faculty id, vehicle info (license plate, make, model, year) and preferred password
2. The user submits the form
3. The system validates the new data
4. If invalid data items, present warning and repeat update attempt
5. Else, save the new data, and display the home form

------------------------

Entry Conditions:

User is NOT registered with system

------------------------

Exit Conditions

User is registered

------------------------

Quality Requirements:

Contact information is provided/verified

=================

=================

Use case name:

Password Reset

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user/driver is presented with a form requesting: new preferred password
2. The user submits the form
3. The system validates the new data
4. If invalid data items, present warning and repeat reset attempt
5. Else, save the new data, and display the home form

------------------------

Entry Conditions:

User is registered with system and has logged in

------------------------

Exit Conditions

User password is changed

------------------------

Quality Requirements:

Valid password provided

=================

=================

Use case name:

Password Reset

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user is presented with a form to input a message and indicate recipient by license plate number
2. The system sends the message to the other driver via established email or text id for matching plate number, else nothing
3. The system indicates the message has been sent

------------------------

Entry Conditions:

User has logged in

------------------------

Exit Conditions

Message sent

------------------------

Quality Requirements:

Valid/ matching plate provided

=================

=================

Use case name:

Logout

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user/driver is presented with a form to confirm that they really want to exit
2. If yes, exit system
3. Else, present prior form

------------------------

Entry Conditions:

User has logged in

------------------------

Exit Conditions

User logged out

------------------------

Quality Requirements:

None

=================

=================

Use case name:

View Campus Map

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user/driver is presented with a form containing the standard UCO map with the handicapped parking lots on campus indicated.

------------------------

Entry Conditions:

User has logged in

------------------------

Exit Conditions

Map displayed

------------------------

Quality Requirements:

None

=================

=================

Use case name:

View Available Spaces

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user/driver is presented with a form containing a list of parking lots from which to select or a campus map with functional "hotspots"
2. Upon selecting a lot, the system will provide a list of handicapped spots with their current statuses and reserved/occupied times

------------------------

Entry Conditions:

User has logged in

------------------------

Exit Conditions

None

------------------------

Quality Requirements:

None

=================

=================

Use case name:

Make Reservation

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user selects a parking space via View Available Spots
2. The user provides intended parking times (Park, Leave) within time limit parameters (e.g. cannot reserve more than 30 minutes in advance, cannot reserve for more than 6 hours, etc)
3. The system marks the space as reserved for that driver

------------------------

Entry Conditions:

User has logged in

------------------------

Exit Conditions

Reservation Completed

------------------------

Quality Requirements:

Request does not exceed rules (30 min in advance, 6 hours total for example)

=================

=================

Use case name:

Change/Extend Reservation

------------------------

Actors:

Driver

------------------------

Flow of events:

1. The user is presented a form with his/her current reserved or occupied parking space with times and status and may indicate a new start time or end time
2. The user may also choose whether to "Cancel" the reservation entirely or to "Leave" the occupied space

------------------------

Entry Conditions:

User has logged in

User has a current reservation

------------------------

Exit Conditions

Reservation Changed

------------------------

Quality Requirements:

Request does not exceed rules (30 min in advance, 6 hours total for example)

=================

=================

Use case name:

Remind User

------------------------

Actors:

Driver

Scheduler

------------------------

Flow of events:

1. The scheduler sends a message to the user to update a reserved space as "Occupied" for each reservation more than 15 minutes past its start time
2. The scheduler sends a message to the user to update an occupied space as "Available" for each space more than 15 minutes past its end time

------------------------

Entry Conditions:

User has logged in

User has a current reservation/occupation

------------------------

Exit Conditions

Message sent

------------------------

Quality Requirements:

None

=================

=================

Use case name:

Navigate to Space

{extends view available space}

------------------------

Actors:

Driver

Google

------------------------

Flow of events:

1. The user views available spaces via the map and selects one
2. The system presents either a stored textual navigation to the spot (from campus entry points) or presents an interactive “map” via the second actor (Google) and passes stored coordinates
   1. The user/driver may interact with the external system (Google Maps)

------------------------

Entry Conditions:

User has logged in

------------------------

Exit Conditions

None

------------------------

Quality Requirements:

None

=================

=================

Use case name:

Clear Stale Reservation

{includes Remind User)

------------------------

Actors:

Scheduler

------------------------

Flow of events:

1. The Scheduler scans existing “Reservations” that are more than 30 minutes past predicted occupation AND “Occupied” spaces more than 30 minutes past departure
2. The system changes these statuses to “Free”

------------------------

Entry Conditions:

User has already been messaged regarding pending space expiration

------------------------

Exit Conditions

Stale reservations cleared

------------------------

Quality Requirements:

None

=================