Team: The Musketeers
Andrew Soth 014248453
Hardit Singh 011635245
Jennifer Nguyen 013012543
Luis Gonzalez (team lead) 014707301
11/1/2018

BUSINESS REQUIREMENTS DOCUMENT

ParkingMaster

Table of Contents

Introduction

The Musketeers is a team of four developers that wants to address an issue that many drivers face every day: finding the best available parking spot. The team aims to create an interactive application that allows users to manage assigned parking in parking lots and structures alike. This application would help streamline the parking process for high-traffic structures and help users find parking in a more timely and efficient manner.

Scope

ParkingMaster will be a mobile-friendly single-page web application with Chrome (Version 70) support that will allow users to search for and reserve spots in designated parking structures. Users can edit their reservations as necessary. Users will also be able to enable notifications for the time remaining on their spot and report other users who have taken their reserved spot.

Only users from North America may register on ParkingMaster.

Personal data may be collected from users for use in telemetry and user access control; this may include:

- Current location and IP address.
- Roles (System admin, admin, management, security, standard) and pertinent role permissions.
- Personal answers to security questions.
- Login, logout, and session length statistics.

Owners of the parking structures must provide detailed maps of the parking structures, including the number of spots in each lot. Administrative accounts will be able to edit parking structure maps and manage user reports.

For our partnered parking structures and lots, the only way to park in the area is through the app, so all of our users who want to park in the structure must have a ParkingMaster account and a device capable of reserving a parking spot. Users must be registered on ParkingMaster to park in these designated parking structures; otherwise, they may be reported and be subject to appropriate consequences.

Stakeholders

Mr. Vatanak Vong accepted the project proposal on September 27, 2018, and will be the main point of contact throughout the development process. Vong will have the final say in any part of the project and will continually give feedback on the existing state of the application after each sprint. We may require contact with Vong during a sprint if urgent issues arise and require an important decision to be made on the state of the project.

Business Completion Criteria

Throughout the life cycle of this project, we will be following agile methodologies and utilizing SCRUM guidelines to provide top satisfaction for our client. Since we are following SCRUM guidelines, we will have at least a completed feature, component, or artifact at the end of each sprint for the client to test and give feedback on. At the end of the 6th sprint, we will have a completed application of ParkingMaster to present and demo.

Assumptions

Drivers must use ParkingMaster to park in lots whose owners have opted into the application. Those without ParkingMaster accounts will not be allowed to park. It is also assumed that users will drive the cars that have been registered under their accounts. Parking with an unregistered car is not allowed, even if the person driving is registered on ParkingMaster.

Drivers who do not follow these rules will be subject to reports and further disciplinary action.

Constraints and budget

This project will undergo two phases - one semester for documentation, and the following semester for implementation. There are approximately nine months from August 2018 to May 2019 to develop and deploy a complete application. Due to this time constraint, the project may be scaled down throughout sprints in order to reach the client's functionality requirements. Since the team is utilizing free technologies during development, time is the only considerable aspect of the budget.

I. Core Functionalities

User Management

1. User Registration

Background/Purpose:
Allows users to register for account with security measures. Allows for baseline interaction with the application.

Actors: All users

User story: A user can register for an account so they can use the application.

Preconditions:
- User is not currently logged in.
- User is on user registration page.
- User has stable internet connection.

Postconditions:
- User information saved to database.

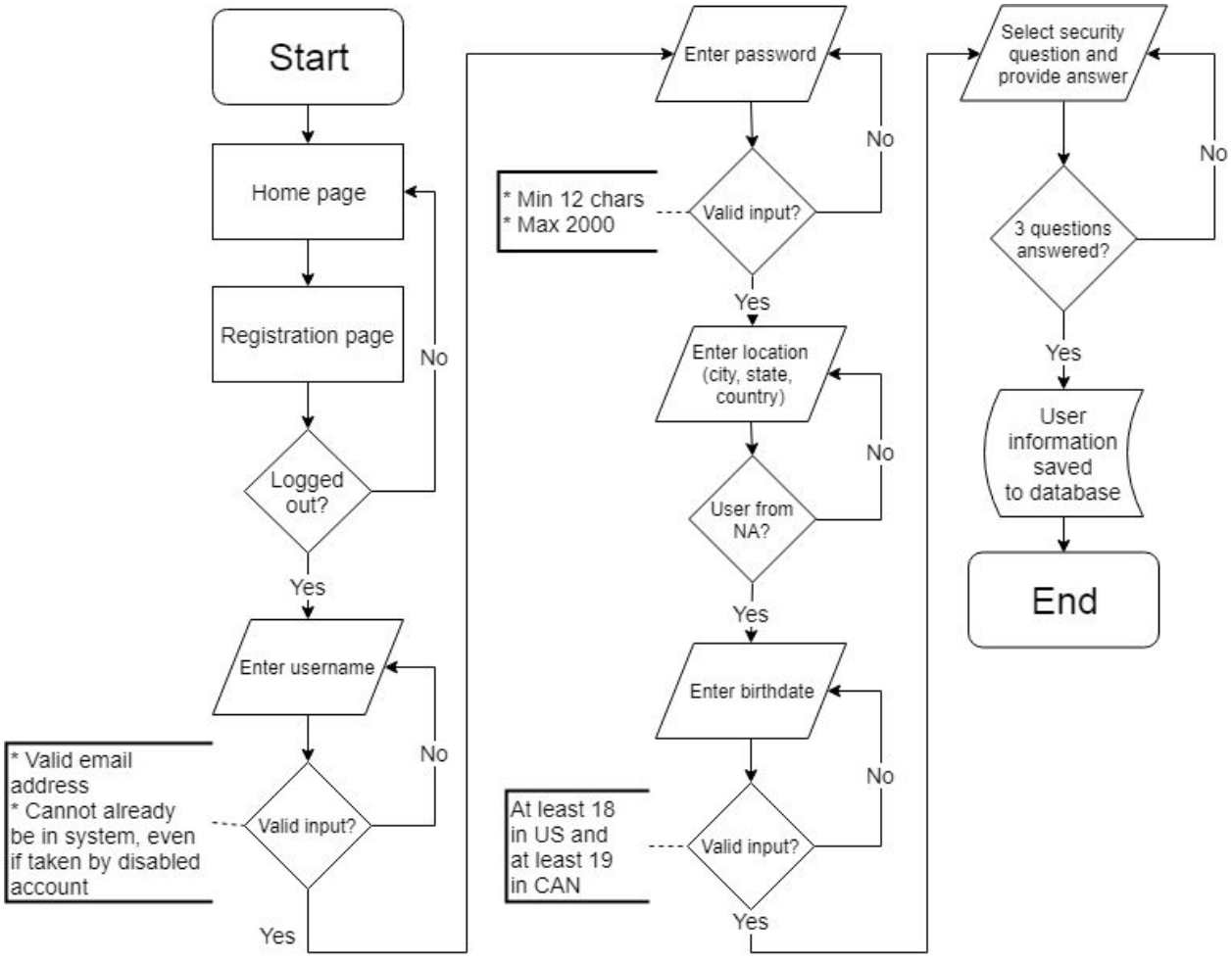
Requirements:
a. User must provide city, state, and country for location.
    i. Users from outside of North America cannot register and will be denied with an error message.
b. User must provide a username.
    i. Username must be a valid email address.
    ii. If the username already belongs to an account in the system, including disabled accounts, the user will be prompted to choose a different username.
    iii. Username cannot be changed after registration.
c. User must provide a password.
    i. Password must be at least 12 characters and at most 2000 characters long.
    ii. Password must adhere to NIST guidelines.
        1. Refrain from repeating or consecutive characters.
        2. Avoid commonly cracked passwords.

iii. All passwords must be protected in the data store.
iv. Brute force attacks on passwords must be unrealistic.
1. An unrealistic attack is defined as taking over 50 years to complete.

d. User must provide birthdate.
i. User must be a legal adult in their country.

e. User must provide answers to 3 security questions in the event that they forget their password.
i. Users may choose security questions from a list of provided choices.

Pass/fail criteria:



2. User Vehicle Registration

Background/Purpose:
Lets users register a vehicle to their account. Vehicle is used to identify user. User is required to register a vehicle before being able to access other parts of application.

Actors: Standard users

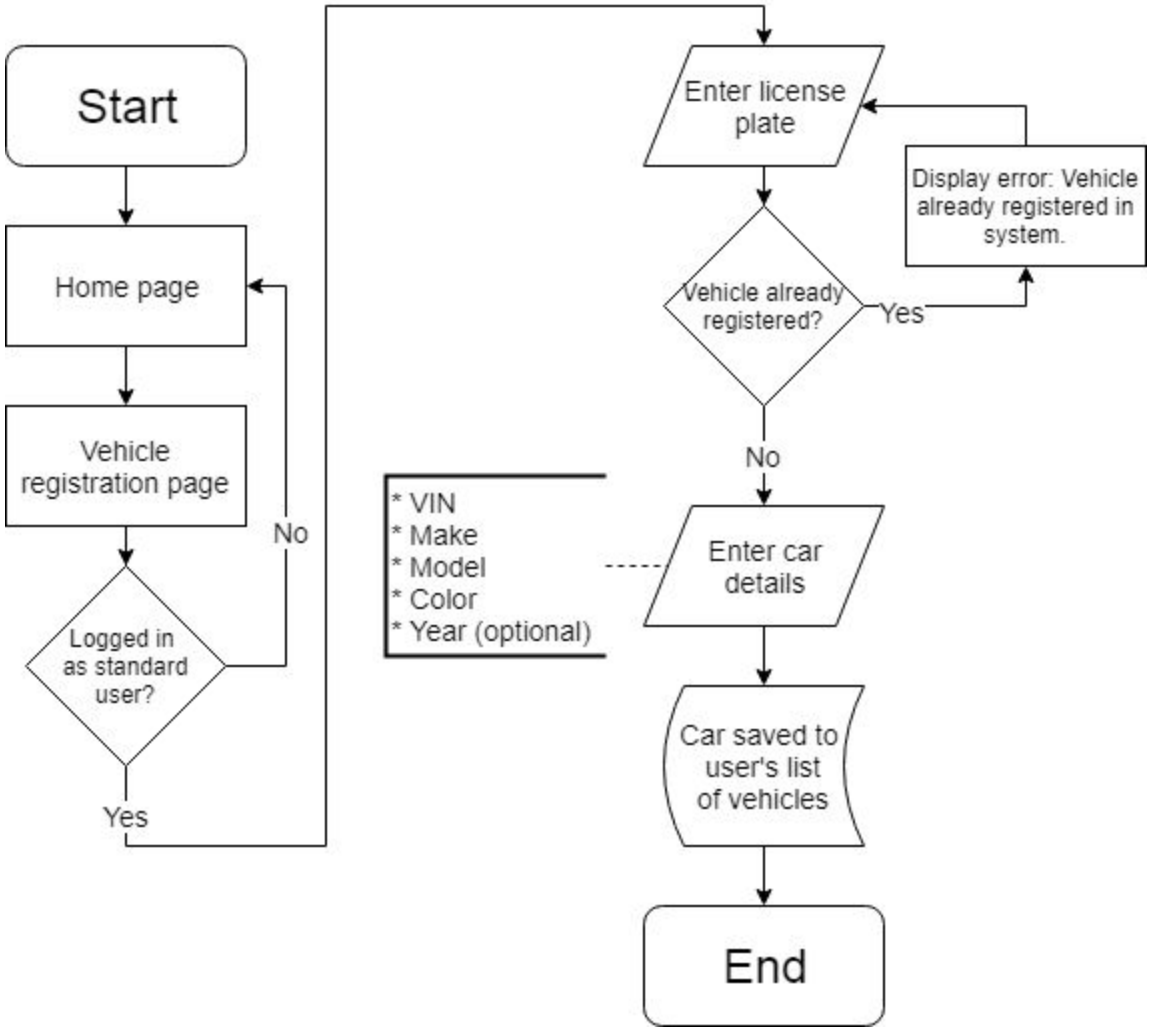
User story: A user can register a vehicle to their account for identification purposes.

Preconditions:
- User is currently logged in.
- User is on vehicle registration page.
- User has stable internet connection.

Postconditions:
- Vehicle is saved to user's registered vehicles.

Requirements:
a. Standard accounts can register new vehicles.
    i. Every vehicle may only be tied to a single account.
    ii. A single account can have multiple vehicles.
b. Vehicle must not be currently in the system.
    i. Vehicle is registered by license plate.
    ii. If user's vehicle is new and has no license plate yet, vehicle will be registered by VIN.
c. User provides vehicle characteristics
    i. VIN
    ii. License Plate
    iii. Make
    iv. Model
    v. Color
    vi. Year (optional)
d. User can delete vehicles that are registered to their account.

Pass/fail criteria:



3. Registration by Higher Level Account

Background/Purpose:
Allows higher level accounts (admin+) to create lower level accounts.

Actors: Admin accounts, management accounts

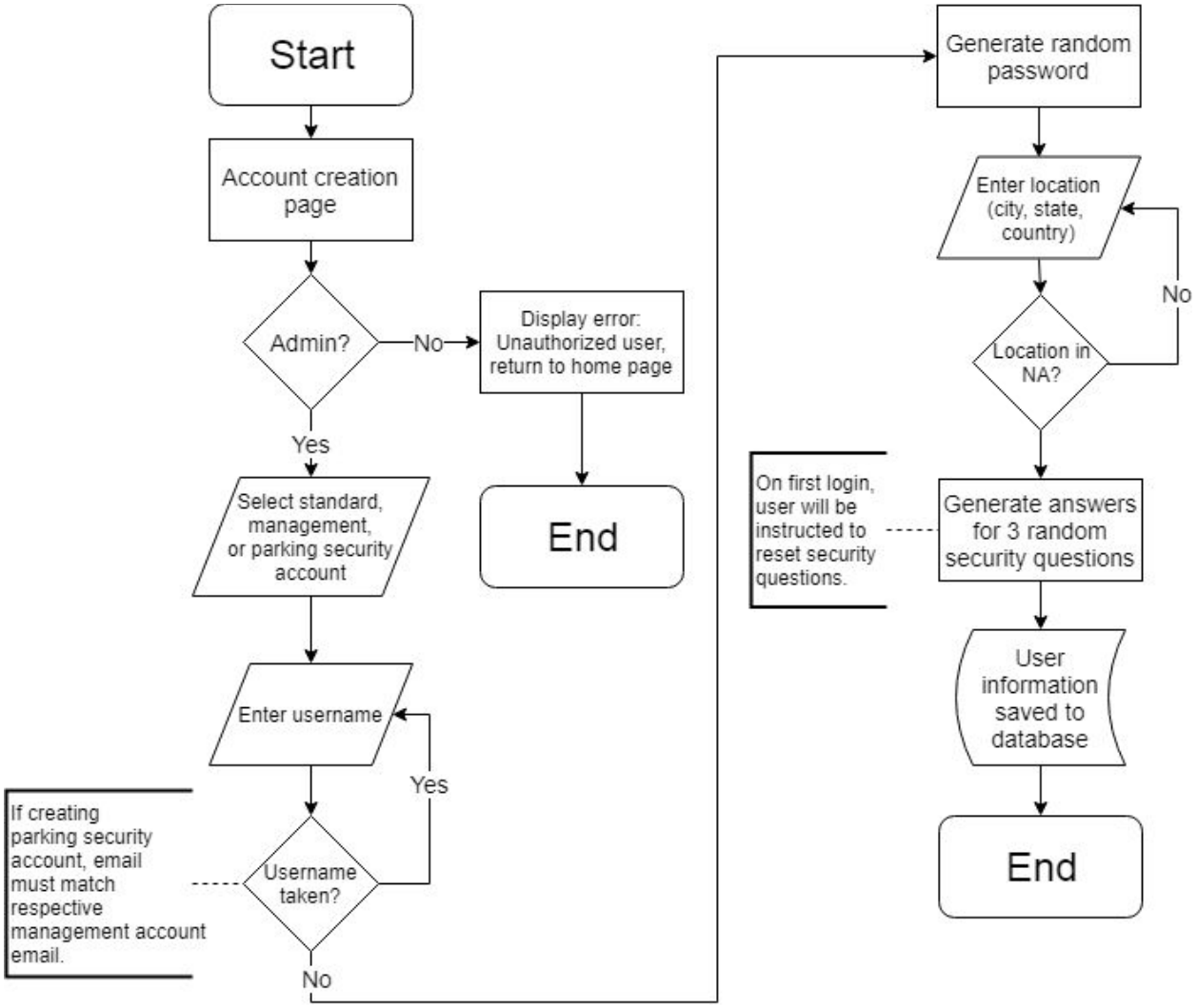
User story: A user with at least admin privileges can create lower level accounts for other users.

Preconditions:
- User is currently logged in.
- User is an administrator or management user.
- User has stable internet connection.

Postconditions:
- User information saved to database.

Requirements:
a. Admin accounts can register standard, management, or parking security accounts.
    i. Admin provides the following account details:
        1. Valid email as username
        2. Date of birth (must be legal adult)
        3. Location within North America (city, state, country)
    ii. If creating a parking security account, the email of their respective management account must be provided.
    iii. Admin accounts are unable to create another admin account.
b. Management accounts can register parking security accounts
    i. The management's email will automatically be filled in when creating a parking security account.
c. The account password is a randomly generated string of 20 digits.
d. The account's security questions are chosen at random.
    i. The answers are filled with a random string of 20 digits.
e. If the username already belongs to an account in the system, including any disabled account, user registration will be denied.
f. On first login:
    i. New password is created.
    ii. New security questions are selected and answered.

Pass/fail criteria:



4. Account activation

Background/Purpose:
Allows higher level accounts to enable or disable lower level accounts.

Actors: Admin accounts, System admin accounts

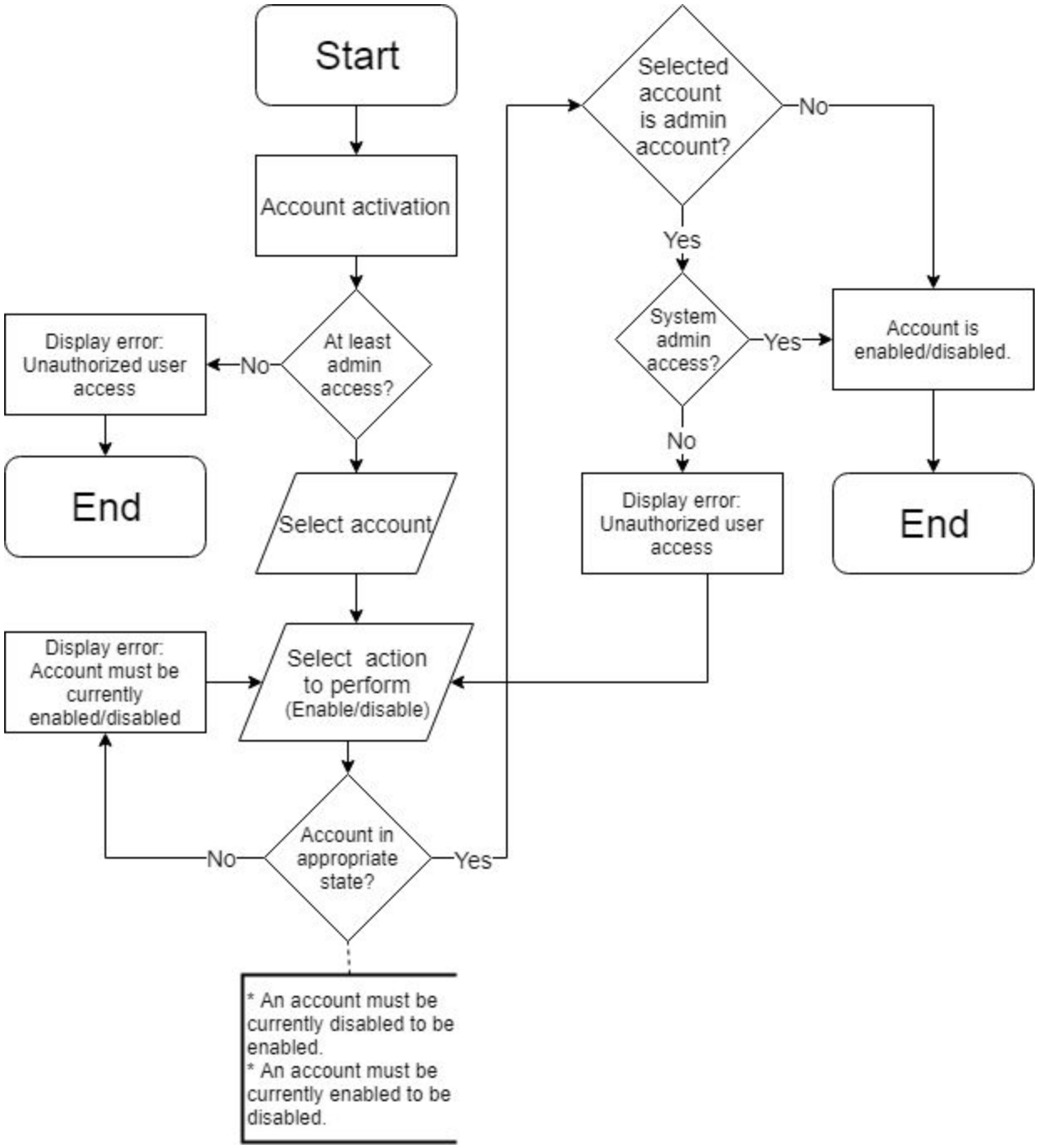
User story: A user with at least admin privileges can enable or disable lower level accounts as necessary.

Preconditions:
- User is currently logged in.
- User is an administrator or system administrator.
- User has stable internet connection.

Postconditions:
- Account appropriately enabled or disabled.

Requirements:
a. Enabling accounts
    i. If an account is currently active, it cannot be activated.
    ii. An admin can activate any account that is below admin status (management, security, or standard account).
    iii. A system admin can activate any account (including other system admin accounts).
b. Disabling accounts
    i. If an account is currently disabled, it cannot be disabled.
    ii. Owners of a disabled account will be notified that their account is disabled and will be unable to login.
    iii. An admin can disable any account that is below admin status (management, security, or standard account).
    iv. A system admin can disable any account (including other system admin accounts).

Pass/fail criteria:



5. Account deletion

Background/Purpose:
Allows users to delete their own accounts and clear the information associated with their account. Also allows users with admin+ permissions to delete other accounts.

Actors: All users

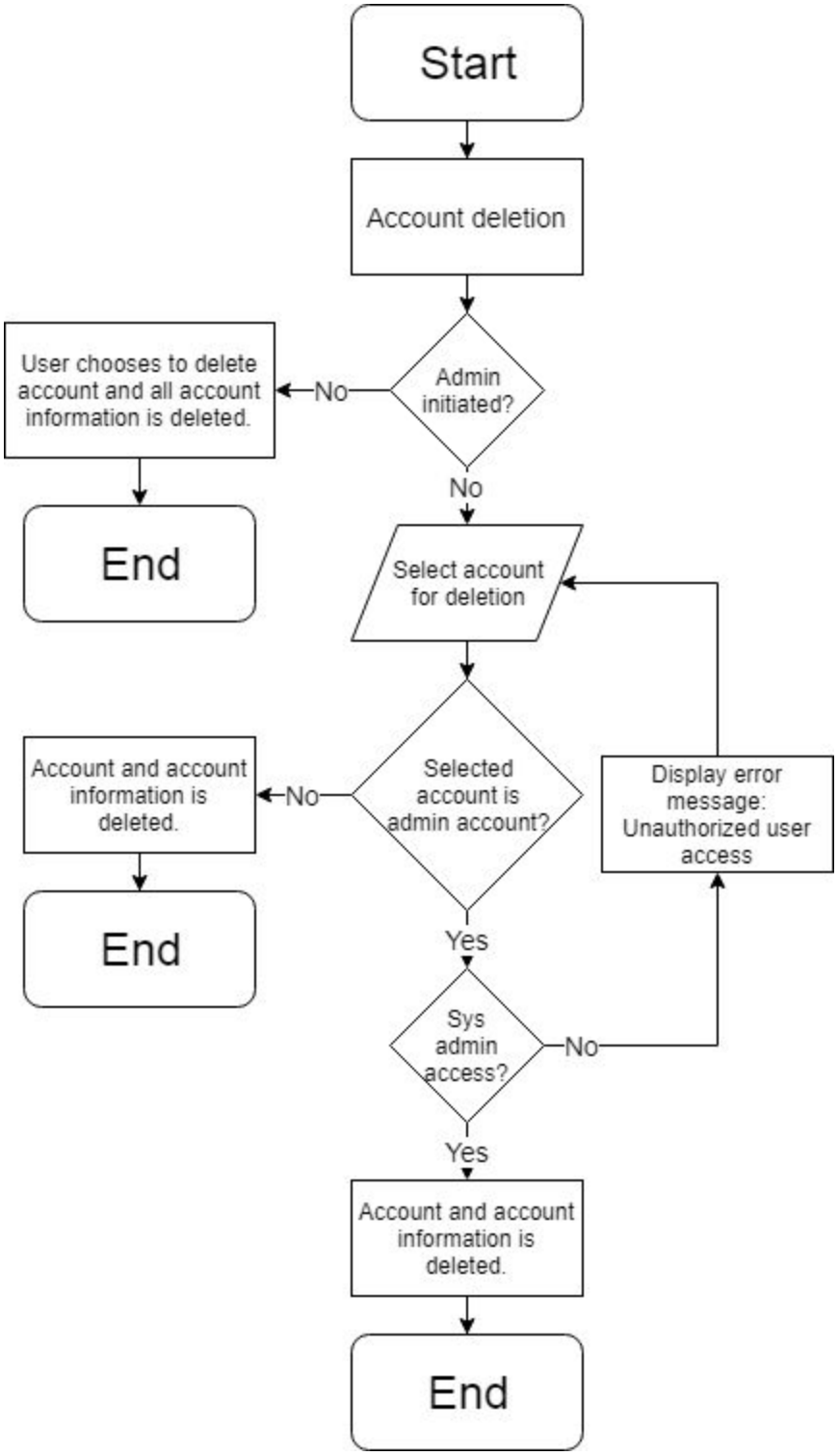
User story:
A user can delete their account along with any pertinent identifying data.
A user with at least admin privileges can delete lower level accounts along with any pertinent identifying data.

Preconditions:
- User is currently logged in.
- User has stable internet connection.

Postconditions:
- Account and all associated information is deleted.

Requirements:
a. User-initiated deletion
    i. A user can delete their own account at any time.
    ii. When a user deletes their account, all of that account's personally identifiable information is deleted.
b. Admin-initiated deletion
    i. An admin can delete any account that is below admin status (management, security, or standard account) at any time.
    ii. A system admin can delete any account (including other system admin accounts) at any time.
    iii. When an admin or system admin deletes an account, all of that account's personally identifiable information is deleted.

Pass/fail criteria:



6. Account configuration

Background/Purpose:
Allows standard users to configure application settings. Also allows admin+ accounts to configure the UAC of lower accounts.

Actors: All users

User story:
A user can configure account settings, including personal account data and privacy settings.

Preconditions:
- User is currently logged in.
- User has stable internet connection.

Postconditions:
- UAC settings saved appropriately.

Requirements:
a. Admin configuration
    i. An admin can change the UAC of any account that is below admin status (management, security, or standard accounts).
    ii. A system admin can change the UAC of any account - this includes other system admin accounts.
b. Application configuration
    i. User may configure the following account settings:
        1. Password
        2. Date of birth
            a. Can cause the user to be denied access to the system if the new DOB means they are not a legal adult.
        3. Location
            a. Can cause the user to be denied access to the system if the new location is outside of North America.
        4. Opt out of telemetry
        5. Delete account/account data

Pass/Fail:
If user changes account information to be out of system scope, they will be banned from the system (ex: Changing age to be under legal adult age, changing location to outside of NA)

Security

7. Login

Background/Purpose:
Lets user login to system and provides remedial action in case user forgets username/password combination.

Actors: All users

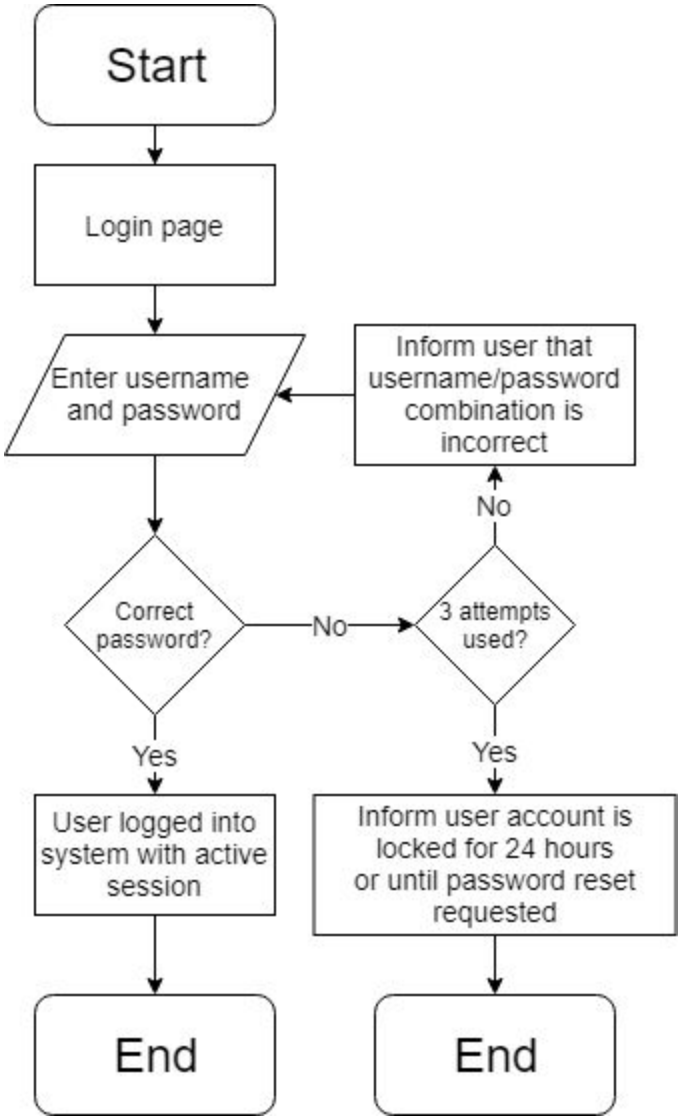
User story: A user can login to the system with the correct username and password.

Preconditions:
- User is not currently logged in.
- User has access to login page.
- User has stable internet connection.

Postconditions:
- User is logged into system with an active session.

Requirements:
a. User must input username and password to login.
b. Account locking
    i. If user fails to login with the correct password after 3 attempts, their account will be locked for 24 hours or until a password reset occurs.
    ii. A locked account can be re-enabled by a user with at least admin privileges (admin, system admin).

Pass/fail criteria:



8. Password reset

Background/Purpose:
Lets user update or reset their password.

Actors: All users

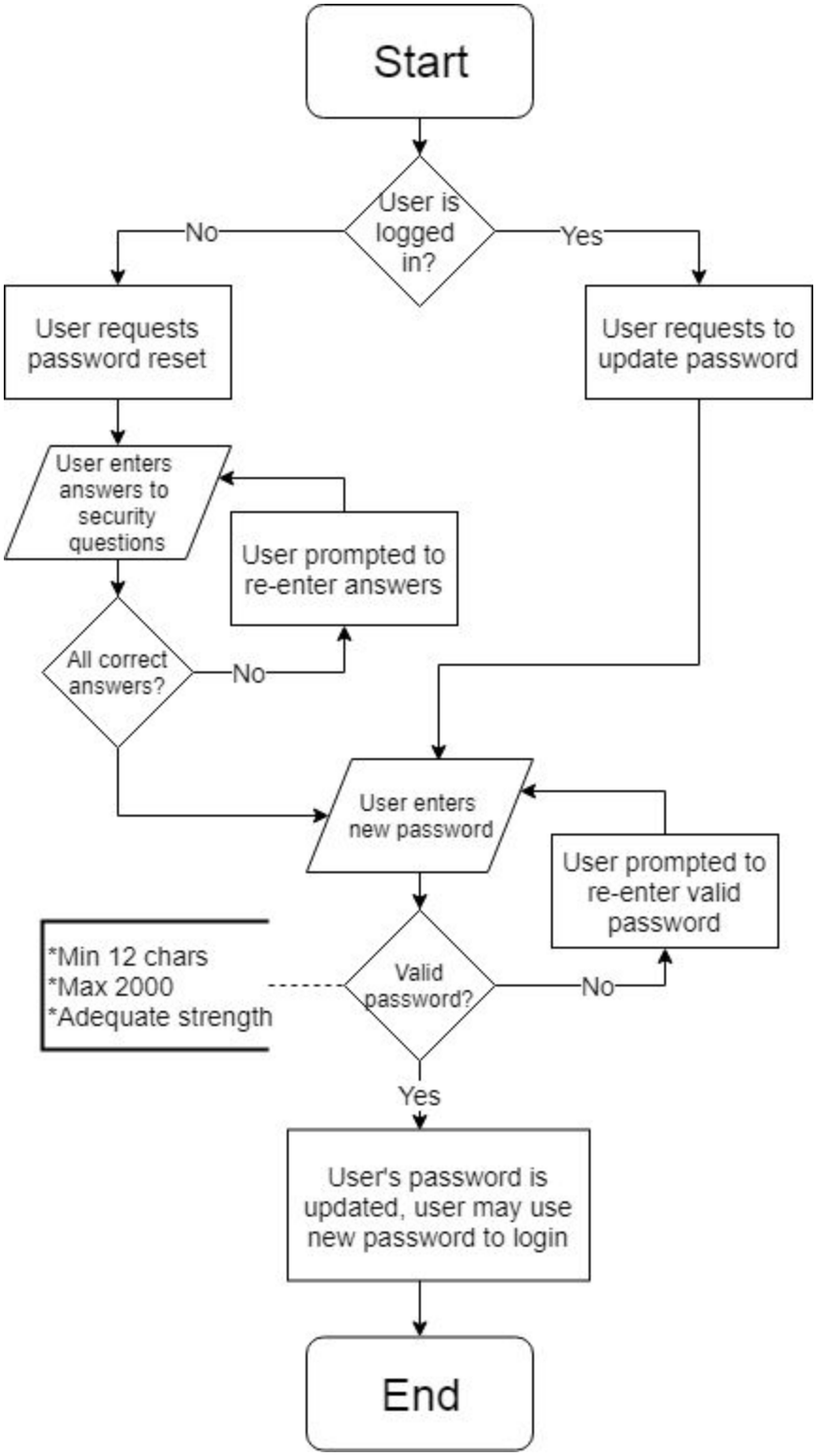
User story:
A user can update their password while logged in.
A user can request a password reset if they are currently logged out.

Preconditions:
- For update - user is currently logged in.
- For reset - user is currently logged out.
- User has stable internet connection.

Postconditions:
- User's password is reset.

Requirements:
a. A user may update their password any time they are logged in.
b. A user must currently be logged out in order to perform a password reset.
c. A user has no limit on how often they can reset their password.
d. A user will be prompted with their security questions when attempting to reset password.
    i. User must correctly answer 3 security questions in order to reset their password and verify a secondary factor when resetting password.
    ii. There is no lockout/penalty if the user fails their security questions.

Pass/fail criteria:



9. Logout

Background/Purpose:
Lets user log out of system.

Actors: All users

User story: A user can log out from the system if they are currently logged in.

Preconditions:
- User is currently logged in.
- User has stable internet connection.

Postconditions:
User is logged out of system.

Requirements:
a. Any currently logged in user can log out of the system.
b. Automatic logout
    i. Occurs if the user is inactive.
        1. A user is defined as inactive if they have not communicated with the server or have not visited another page in the last 30 minutes.
    ii. Once a user performs an action that communicates with the server or visits another page, their session will be extended.

Pass/fail criteria:
If a user is not logged into the system, they cannot log out.

10. Authorization / User access control

Background/Purpose:
Restricts the content that a user has access to, based on their role.

Actors: All users

User story:
A user can have access to various parts of the system depending on the privileges they have.

Preconditions:
- User is registered in system.
- User is logged in.

Postconditions:
User is allowed access appropriately.

Requirements:
a. Users should have appropriate access to the system based on their role privileges.
b. UAC validation will occur for users who are currently logged in.
c. Users will have the ability to configure security settings for content and functionalities.
d. To access the system, users must be
    i. Registered.
    ii. From North America (the base scope).
    iii. Legal adults in their country.
e. Unauthorized users will be informed with an error message and redirected to the appropriate page.
    i. A server request by an unauthorized user will return an error.

Pass/fail criteria:
If user does not have sufficient privileges to access a functionality, an error will inform them they do not have access, and they will be returned to the previous page.

11. User privacy

Background/Purpose:
Sets guidelines and End-User License Agreement guidelines that must be accepted.

Actors: Admin, Security, Standard users

User story:
A user must the accept the current EULA to access application.
A user can configure privacy settings for the application.

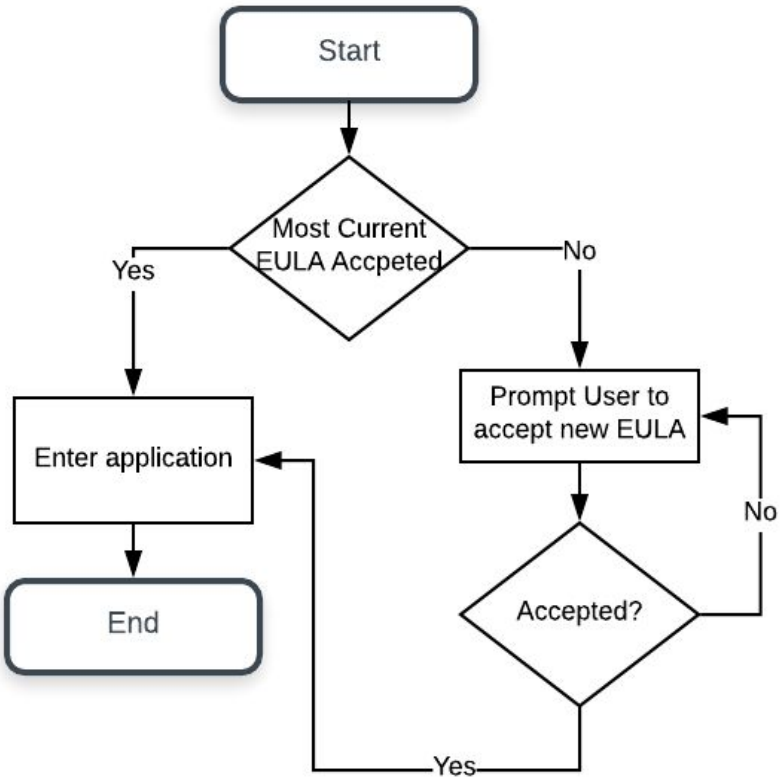
Preconditions:
- User has logged in.
- User falls into one of two categories:
    - Has not yet accepted new EULA
    - Has never accepted the current EULA beforehand

Postconditions:
- Once EULA accepted, user may access system normally - account updated within database as having accepted most current EULA.
- Privacy settings saved appropriately.
- If user decides to delete data, data deleted from system appropriately.

Requirements:
a. Terms of Service / End-User License Agreement (EULA)
    i. System admins can manage EULA by:
        1. Adding one.
        2. Updating one.
        3. Deleting one.
    ii. One and only one EULA may be active.
    iii. Users must accept the most current EULA to have access to application functions.
b. User data privacy
    i. Personal information and telemetry data can only be collected from users who have explicitly opted into data collection.
    ii. Users will have the option to opt out of personal data collection.
    iii. Users may see what personal data is being collected and stored

iv. All user data is deleted when a user deletes their account.
    1. Users will have the option to delete their user data without deleting their account.

Pass/fail criteria:



Error Handling

12. Exception handling

Background/Purpose:
Provides mechanism to identify exceptions and reroute user appropriately.

Actors: All users

User story: A user is shown an error message and remedial action whenever they encounter an exception.

Preconditions:

- User has access to the system.

Postconditions:

- User is shown an error message and remedial action.

Requirements:

a. A user friendly message will specify type of error and whether the error occurred on the client-side or server-side.
    i. No server response.
    ii. Request has resulted in error.
    iii. Internal server error.
    iv. Input validation has failed.
    v. User does not have permission to access feature.
b. A remedial message is always shown to the user after an exception occurs.
    i. Appropriate course of action.
    ii. System admin should be notified.
c. Exceptions should not bring down the entire system, except during situations such as a server or network shutdown.

Pass/fail criteria:
Exceptions should always be logged with pertinent information.
Users should always be shown a message with exception information and appropriate course of action.

Audit management

13. Logging

Background/Purpose:
Provide detailed logging for errors and telemetry, as well as monitoring against malicious attacks.

Actors: All users

User story:
User activity monitored and stored in database.
User submissions stored in database.
Server activity monitored and stored in database.

Preconditions:
- Server is pinged.

Postconditions:
- Everytime server is pinged, reason for request in logged, data sent/received is logged, function executed is logged, or error is logged.

Requirements:
a. Error logging
    i. System admin interactions
        1. Notified if 100+ error logs fail
        2. Can delete error logs - no other account type has this permission
    ii. All exceptions for all users are logged with the following:
        1. Timestamp (Date and time)
        2. Error message displayed to the user
        3. Internal errors sent to System Admin
        4. Line of code where error occurred logged
        5. User that experienced the exception
        6. Request or action that resulted in exception
b. Telemetry
    i. System admin interactions
        1. Notified if 100+ telemetry logs fail

ii. For all users except for those who have opted out of telemetry, the following data will be logged:
    1. Date and time of:
        a. Login
        b. Logout
        c. Page visit
        d. Functionality usage
    2. IP address
    3. Location

c. Malicious attacks
    i. System will monitor for denial of service (DOS) attacks
        1. Log all requests to the server
        2. Monitor logs for suspicious activity

Pass/fail criteria:

If a substantial number of logs fail (100+), the system admin must be notified.

Otherwise, logs should always be recorded with pertinent information.

14. Archiving

Background/Purpose:
Provides archiving mechanism for logs.

Actors: All users

User story: N/A

Preconditions:
- Logging has occurred for at least 30 days.

Postconditions:
- Logs are grouped, archived, or deleted appropriately.

Requirements:
a. Back-up
    i. Group and archive
        1. Logs 30+ days old
        2. Logs 2+ years old
    ii. Delete
        1. Logs 5+ years old
    iii. In case archiving fails
        1. Retry every 2 hours
        2. After 3 failed attempts, stop archiving retries and notify the system admin.

Pass/fail criteria:
If archiving attempts fail several times (3+ attempts), a system admin must be notified.
Otherwise, logs should be grouped and archived appropriately based on their age.

System Analytics

15. Usage analysis dashboard

Background/Purpose:
Visually represents user metrics in graph form.

Actors: All users

User story:
A user performs an action that creates data that can be analyzed and represented.

Preconditions:
- User performs an action.

Postconditions:
- Data is collected and represented.

Requirements:
a. The following metrics are collected and represented as bar charts:
    i. Average logins per month vs. Number of currently registered users
        1. Max and min bars shown
    ii. Average length of session per month
        1. Max and min bars shown
    iii. Login attempts
        1. Failed vs. Successful
    iv. Top 5
        1. Average time spent on any page
        2. Most used system features
b. The following metrics are collected and represented as line charts:
    i. By month over the last 6 months
        1. Average session duration
        2. Number of logged in users

Pass/fail criteria:
Any valid and pertinent data should be collected and represented visually.

Data Store Access

16. Data access layer

Background/Purpose:
Provide framework for Data Access Layer (DAL) create, read, update, and delete (CRUD) operations.

Actors: All users

User story:
A user requests access to some data.
A user inputs data that needs to be stored.
A user edits data that has already been stored.
A user deletes data from store.

Preconditions:
- Creating: data store should not have any data in directed field.
- Reading: Data can only be read by authorized user. Data must exist.
- Updating: Data can only be updated by authorized user. Data must exist.
- Deleting: Data can only be deleted by authorized user. Data must Exist

Postconditions:
- Appropriate action performed on records and saved to data store.
- All actions logged.

Requirements:
a. Writing records
    i. New records must be able to be added to the data store.
    ii. Attempting to add a duplicate of an existing entry will result in an error.
    iii. Nothing will occur if there is not enough space in the data store to store a new record.
    iv. Operations must ensure the data store remains in a consistent state.
b. Reading records
    i. Must be able to read all data from data store or just a specific subset of data.

ii. Nothing will occur if specified record is not in data store.
iii. Operations must ensure the data store remains in a consistent state.

c. Updating records
    i. Existing records must have the ability to be updated.
    ii. Nothing will occur if specified record is not in data store.
    iii. Operations must ensure the data store remains in a consistent state.
d. Deleting records
    i. Existing records must be able to be deleted.
    ii. Nothing will occur if specified record is not in data store.
    iii. Operations must ensure the data store remains in a consistent state.

17. Data restriction
    a. Data access restricted based on function and authorization.

Pass/fail criteria:
Adding duplicate record will cause error.
Adding requires proper authorization dependent on type of data being added.
Reading, updating, or deleting a record that does not exist will cause an error.
Adding record will pass if user is authorized and record does not already exist.
Reading, updating or deleting a record will pass if user is authorized.

Documentation

18. User manual

Background/Purpose:
Provide documentation that any users may access without authorization.

Actors: All users

User story: A user can view developer documentation, the user manual, or the FAQ for the application.

Preconditions:

- User has stable internet connection.

Postconditions:

- User can view requested documentation.

Requirements:

a. Developer documentation
    i. Documentation will be available for software developers who wish to maintain or continue development of the system.
    ii. Authentication is not necessary to view developer documentation.
b. User manual
    i. An exhaustive user manual will be available for normal users of the system.
    ii. Authentication is not necessary to view the user manual.
    iii. Updates are handled by a system admin.
c. Frequently Asked Questions (FAQ)
    i. A non-exhaustive FAQ will be available for normal users of the system.
    ii. Authentication is not necessary to view the FAQ.
    iii. Updates are handled by a system admin.

Pass/fail criteria:
As long as user has stable internet connection, and documentation is currently available, they will be able to view the requested documentation.

II. Unique functionalities

Reservation System

Background/Purpose:
Allows standard users to make, edit, and cancel their reservations.

Actors: Standard users

User story:
A user can make a reservation for their parking spot.
A user can edit or delete their reservation for their parking spot.

Preconditions:

- User is currently logged in.
- User has stable internet connection.
- User is on reservation page.

Postconditions:

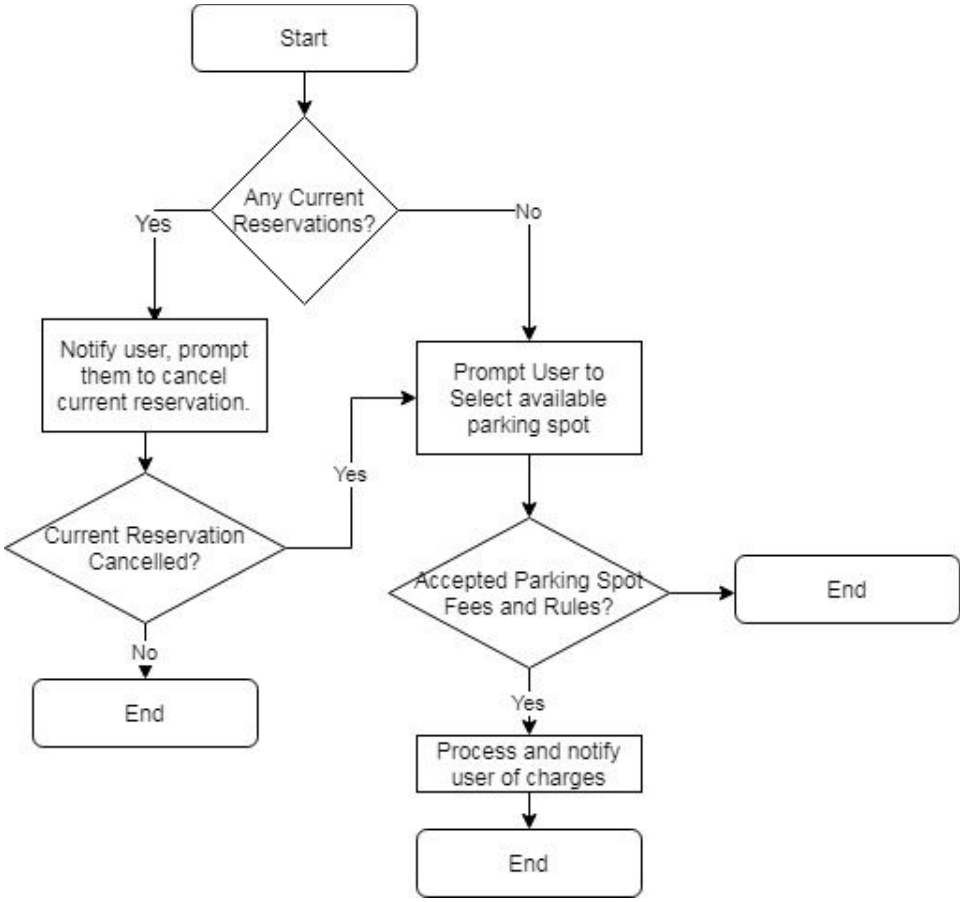
- Reservation data is saved.

Requirements:

19. View Parking Lot Map
    a. Image viewer
        i. Ability to zoom in.
        ii. Scrollable when zoomed in.

20. Reserving a Spot
    a. Select an available parking spot from a list
        i. List is scrollable.
        ii. All unavailable parking spots are greyed out.
    b. Select reservation duration
        i. Spots are reserved by a timer by selecting how many hours and minutes the reservation will last.
        ii. Reservations cannot be longer than the parking lot specified maximum duration.
        iii. If the user enters a timer longer than the parking lot maximum, the reservation will fail to go through and the user will be notified.
        iv. If the user is not able to cover the cost of the reservation with their current balance, as described in Payment System, the reservation will fail.

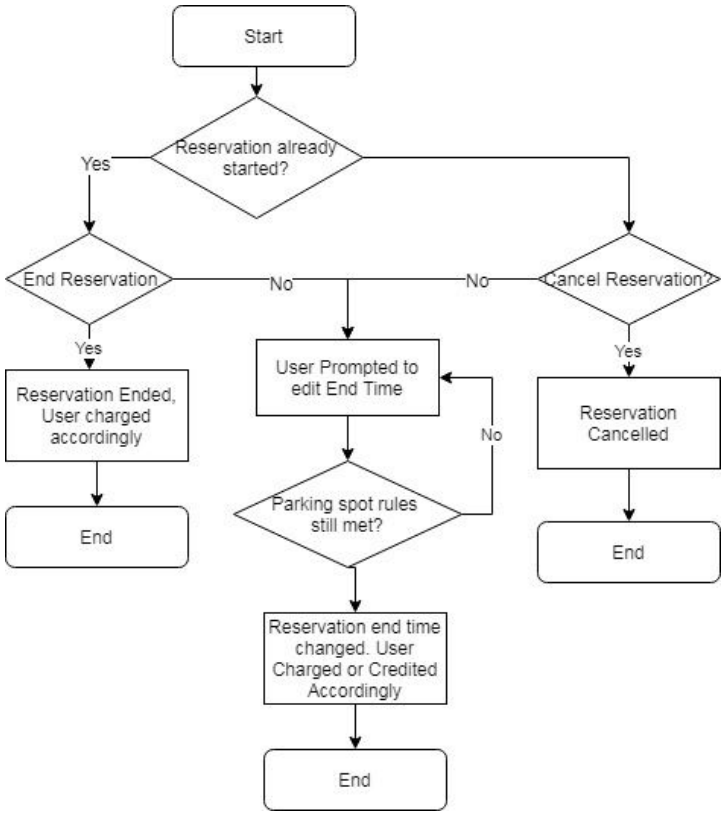
Pass/fail criteria:



21. Editing Reservations

a. Extending
    i. A reservation may be extended at any point.
    ii. A user selects how many hours and minutes they wish to add.
    iii. A single reservation cannot be extended past the parking lot 3.specified maximum duration.
    iv. If an extension would pass the parking lot specified maximum, the extension will not go through and the user will be notified.
    v. If the user is unable to cover the extension with their current balance, the extension will not go through.
b. Shortening
    i. A reservation may be shortened at any point.
    ii. A user selects how many hours and minutes they wish to remove.
c. Canceling
    i. A reservation may be canceled at any point.
    ii. Canceling a reservation sets remaining time to 0.

Pass/fail criteria:



Lot Management

Background/Purpose:
Allows parking manager accounts to provide and edit parking lot data.

Actors: Parking manager users

User story:
A parking manager user can provide and edit data for their parking lots.

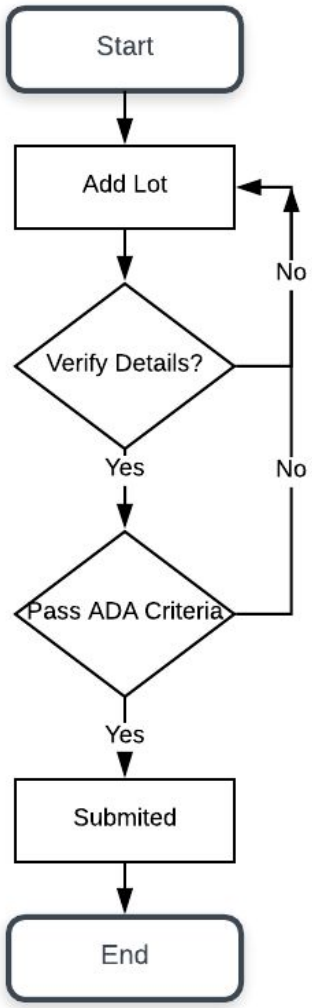
Preconditions:
- User must be logged in as a parking manager.
- User must have stable internet connection.
- User is on lot management page.

Postconditions:
- Parking lot data is saved.

Requirements:

22. Adding Lots
    a. Parking lot data file upload.
        i. Data includes the total number of parking spots
        ii. The total number of handicap parking spots
            1. If lot fails to comply to ADA requirements, lot will be rejected.
        iii. The name/label for every parking spot.
    b. Image upload
        i. Image should include entire parking lot.

Pass/Fail criteria:

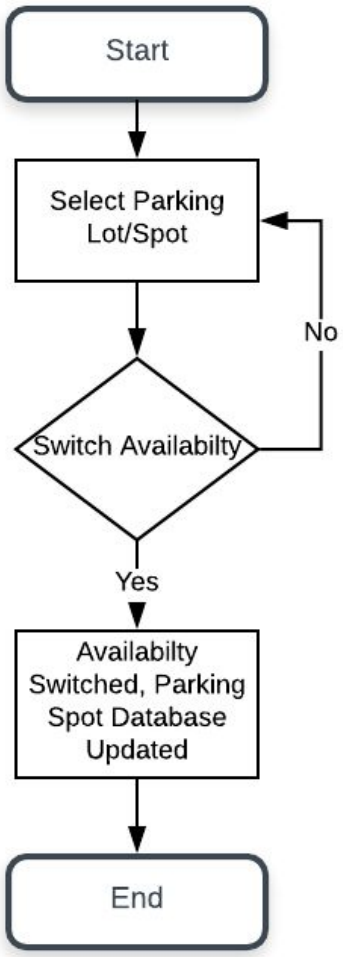


23. Editing Lots
    a. Image upload for new map.
        i. Overwrites the previously saved parking lot map.
        ii. Clearly warns that the old map will be overwritten.
    b. Set lot name
        i. Maximum of 50 characters
    c. Set parking rate
        i. Rate is described in dollars per hour.
    d. Set parking hours
        i. Set opening parking hour
        ii. Set closing parking hour
        iii. If parking hours are not set, the parking lot is assumed to never close.
    e. Set maximum reservation time
        i. If maximum parking length is not set, then there is no limit to a reservation length.

Pass/fail criteria:

24. Editing Parking Spots
    a. Disable a parking spot to set it to unavailable
    b. Re-enable a parking spot to available
    c. Parking space viewer
        i. List view of which parking spots are currently enabled or disabled.
        ii. Ability to enable/disable parking spots from this view.
        iii. View up to 50 parking spaces per page.
    d. Re-verify ADA standards
        i. Every time the parking lot spaces are edited.
        ii. If the change fails this check, the edit is denied.

Pass/fail criteria:



Abuse Management

Background/Purpose:
Allows standard accounts to submit abuse reports.
Allows management accounts to view and verify abuse reports and discipline users appropriately.

Actors: All users

User story:
A standard user can submit an abuse report when another user performs inappropriate behavior.
A management user can process abuse reports to provide disciplinary action.

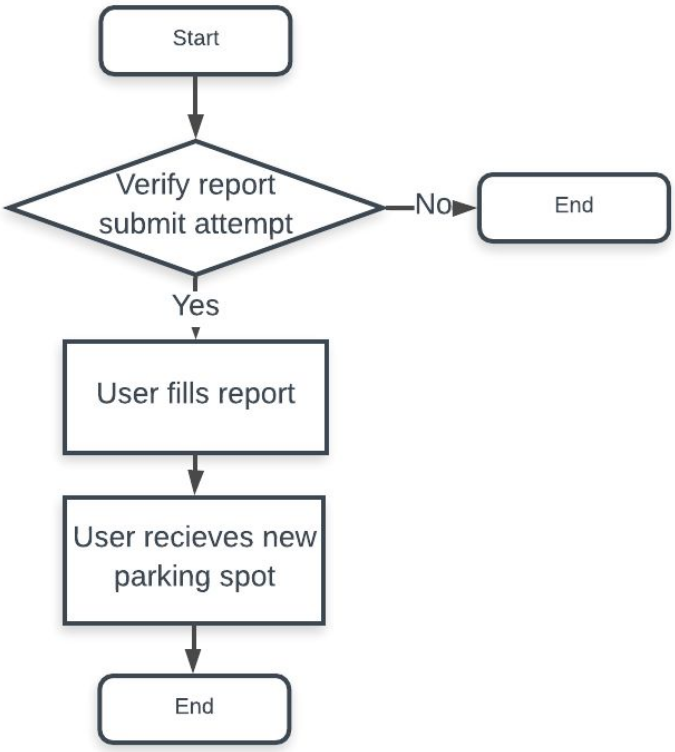
Preconditions:
- User is currently logged in.
- User has stable internet connection.
- Standard user: Can only submit abuse report for the spot they have reserved.

Postconditions:
- Abuse report is sent and handled.

Requirements:

25. Submitting Reports
    a. Button on reserved parking space page
        i. A user can only report a vehicle parked in their reserved spot.
        ii. After report is submitted, their reservation is canceled allowing them to reserve a new spot.
    b. Upload photo of car
        i. Maximum 10 MBs
        ii. File upload completes within 2-20 seconds with a 4G mobile connection.
        iii. Include timestamp.
    c. Upload License Plate/VIN
        i. Manually input by the user
            1. Maximum 8 characters for the license plate and/o VIN
            2. 2 characters for the state

Pass/fail criteria:



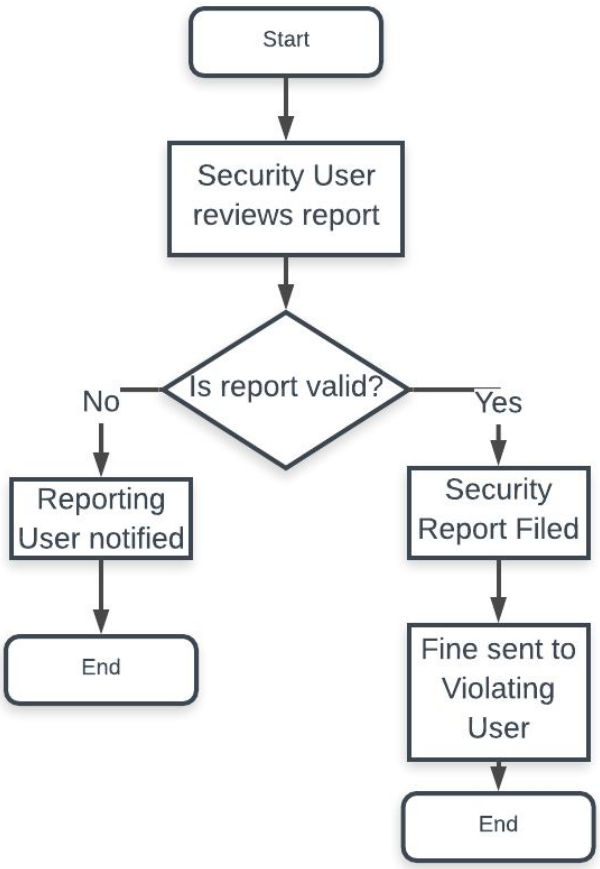
26. Viewing Reports
    a. Reports presented in a list
        i. Most recent reports are highest on the list
        ii. Show up to 20 reports per page
        iii. Only management accounts may view reports for their own parking lot

27. Verifying Reports
    a. Management accounts and their parking security accounts verify all reports.
    b. Upload photo of car
        i. Maximum 10 MBs
        ii. File upload completes within 2-30 seconds with a 4G mobile connection.
        iii. Include timestamp.

iv. Saved along with the original evidence provided in the original report.

28. Fines and bans
    a. User search by license plate/VIN
        i. Only management and admin accounts have this functionality.
        ii. Search only returns user email and information on past infractions
    b. Ticket the car owner's user account if registered
        i. Customizable amount by the management account
        ii. Maximum fine amount: $100
        iii. User cannot reserve another spot until the fine is paid
    c. Ticket the license plate if not registered
        i. Any future account claiming the car has the ticket applied to them.
    d. Ban a user
        i. User will no longer be allowed to reserve parking spots in this particular lot.

Pass/fail criteria:



Payment System

29. Balance System

Background/Purpose:

Actors: All users

User story: A user can add money to their account to pay for reservations or fines.

Preconditions:

- User is currently logged in.
- User has stable internet connection.

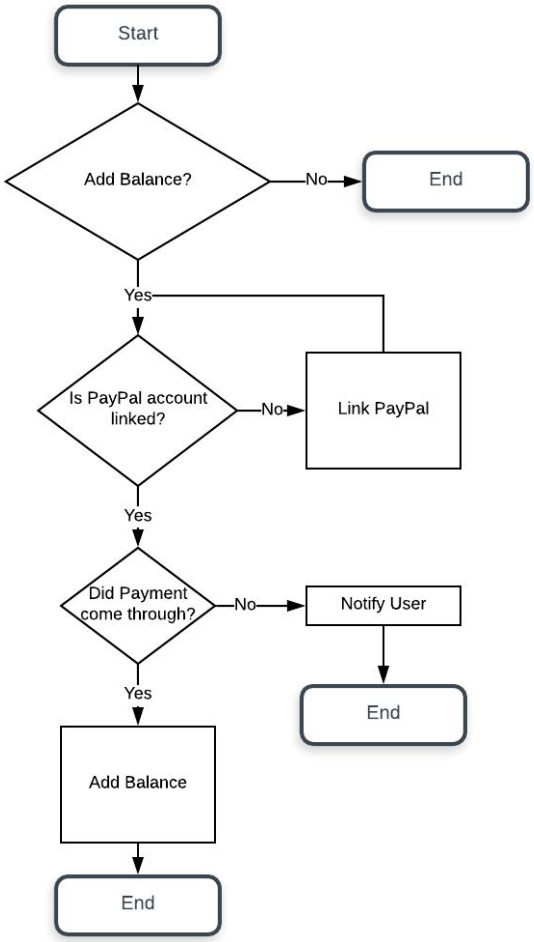
Postconditions:

- Credits or Debits will be updated in server as soon as processes function.

Requirements:

a. Add balance
    i. A user must pay upfront to add money to their account, stored as a balance.
b. Use balance
    i. All charges from reservations or fines are taken from this balance.
    ii. Only fines can bring the balance to a negative.

Pass/fail criteria:



30. Charging system

Background/Purpose:

Actors: Standard User, Security User

User story: Standard users account is charged when user reserves a parking spot. Standard user is charged when user receives a citation from a Security User.

Preconditions:

- User is registered.

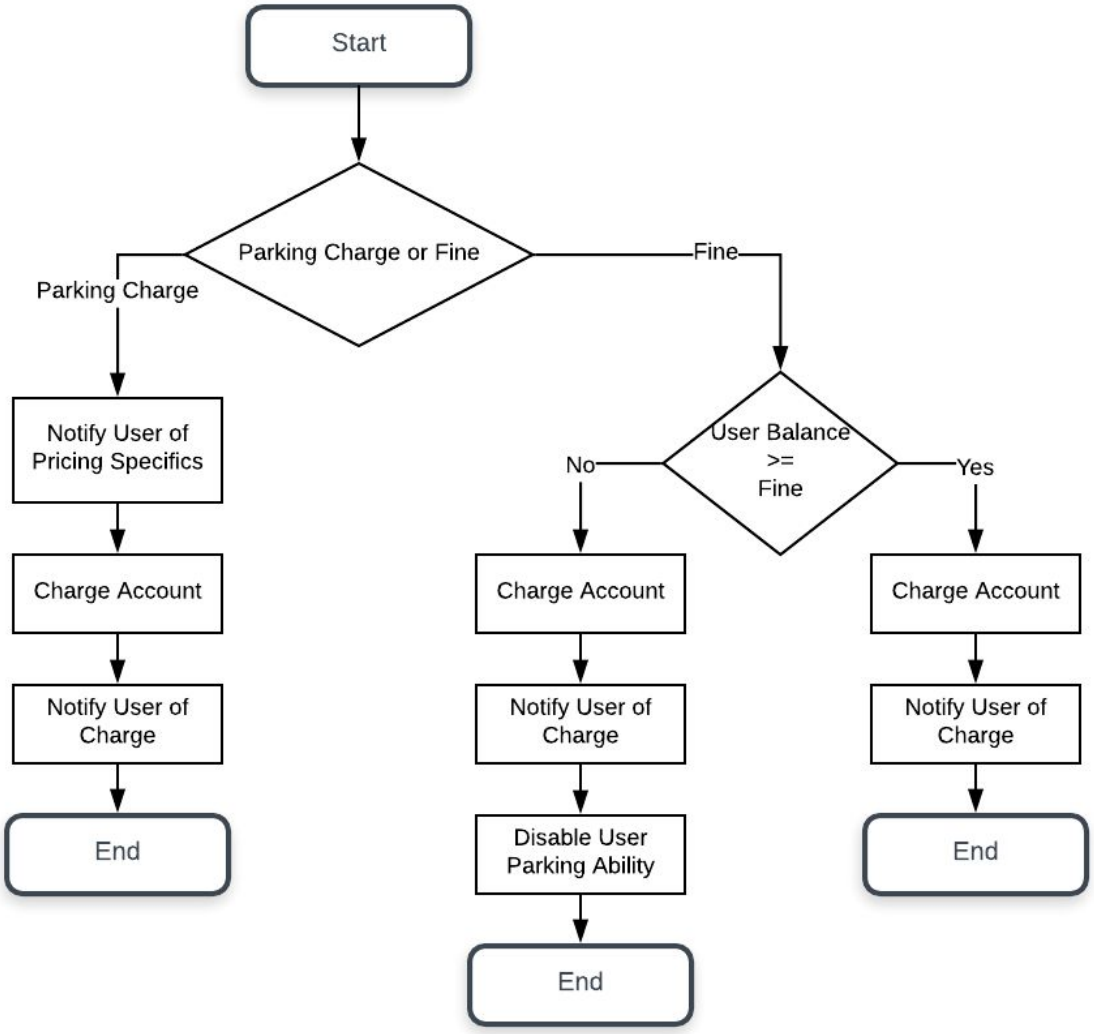
Postconditions:

- Charged account is deducted from balance.
- Account is updated to reflect new balance immediately.

Requirements:

a. Tracking bills
    i. User maintains balance from automatic and manual charging.
b. Automatic charging
    i. User will be automatically charged for the duration of their parking at the end of their reservation.
c. Manual charging
    i. User will be charged manually by a management account if they are fined.

Pass/fail criteria:



Notifications

31. Notification center

Background/Purpose:
Allows users to toggle notifications.

Actors: All users

User story: User enables/disables specific notification.

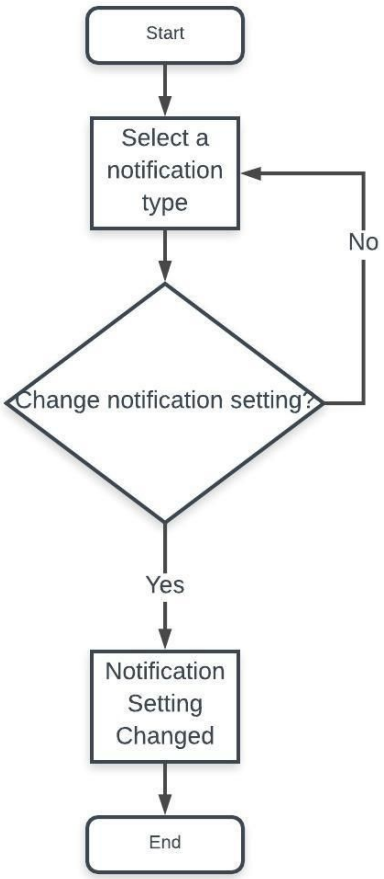
Preconditions:
- User is currently logged in.
- User has stable internet connection.

Postconditions:
- Notification settings saved.

Requirements:
a. Opt in/ Opt out
    i. All notifications can enabled or disabled in the Notification center on an individual basis.
    ii. By default, all notifications are off.
b. Admin Notifications
    i. Notified every time a management account registers. Precondition: management account submitted registration.
c. Management Account Notifications
    i. Notified when reports are received in their lot.
d. Parking Security Account Notifications
    i. Notified when reports are received in their lot.
e. Standard Account Notifications
    i. User notified when they are fined.
    ii. Receive notifications on the status of favorite spots from favorites list at user specified times.
    iii. User notified when they are banned from a lot.
f. Notification viewer.
    i. All notifications are presented in a list view.
    ii. Up to 20 notifications are selectable per page.

iii. The notifications icon has current number of unviewed notifications partially overlapping it.

Pass/fail criteria:



Favorites List

Background/Purpose:
Allows users to add, delete, view, or manage a Favorite Parking lot list.

Actors: Standard user

User story:

- User adds Favorite parking lot to Favorites List.
- User adds favorite parking spot to favorited parking lot.
- User deletes favorite parking lot from favorites list.
- User deletes favorite parking spot from favorited parking lot.
- User views all favorited parking lots.
- User views favorite parking spots in favorited parking lot.

Preconditions:

- User is currently logged in.

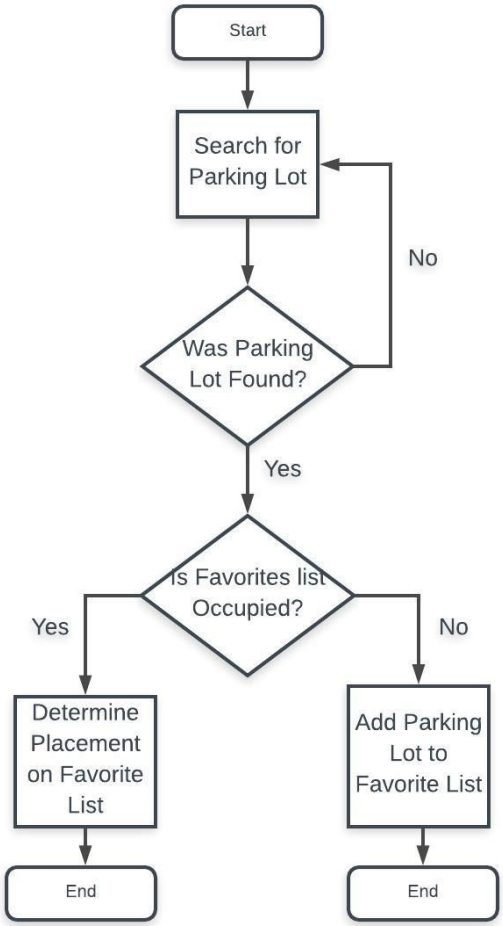
Postconditions:

- Favorites are saved appropriately depending on operation.

Requirements:

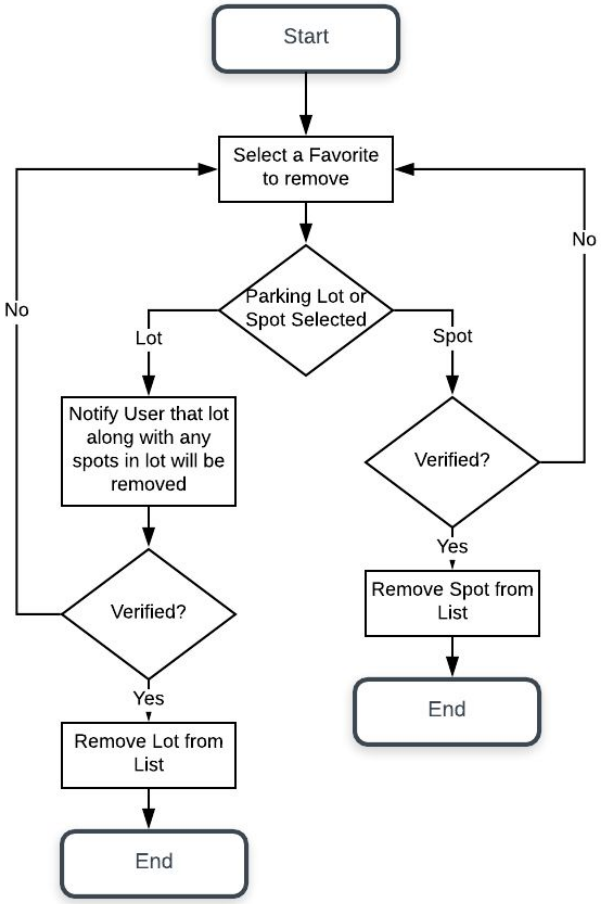
32. Managing favorites
    a. Adding Favorites
        i. Every lot has a button that can be pressed to be added to the user's favorites.

Pass/fail criteria:



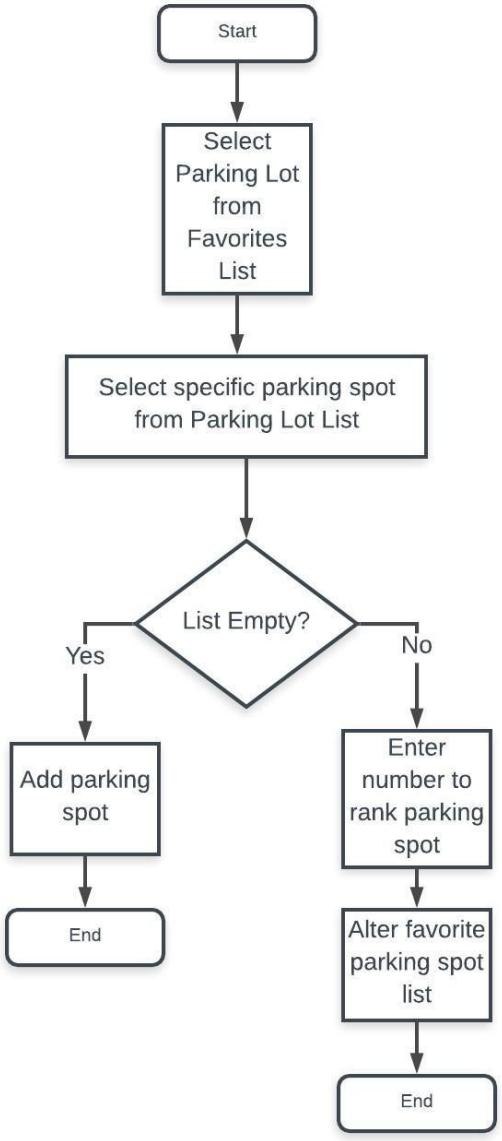
b. Reordering Favorites
    i. User can reorder lots in their Favorites list.
c. Removing Favorites
    i. User can remove lots from their Favorites list.

Pass/fail criteria:



33. Favorite parking lots/spots
    a. Selecting favorite parking spots
        i. User selects spots from a list view by “checking” a spot they wish to add.
        ii. Specific notifications can be set up for these parking spots.

Pass/fail criteria:



34. Viewing Favorites
    a. Select Favorite from list of Favorites
        i. Favorited parking lots are displayed in page form.
        ii. Each page shows up to 5 parking lots at a time.

Pass/fail criteria: