# Big Blue Parking Genie

Team 5

-Anne Jetton: Team Manager

-Makalee Beelek: Graphic Designer

-Derek Ward: Back-End Specialist

-Kate Sergent: Front-End Specialist

# Project Summary

Description

- The Big Blue Parking Genie is a web application that allows users to purchase and rent out parking spots.

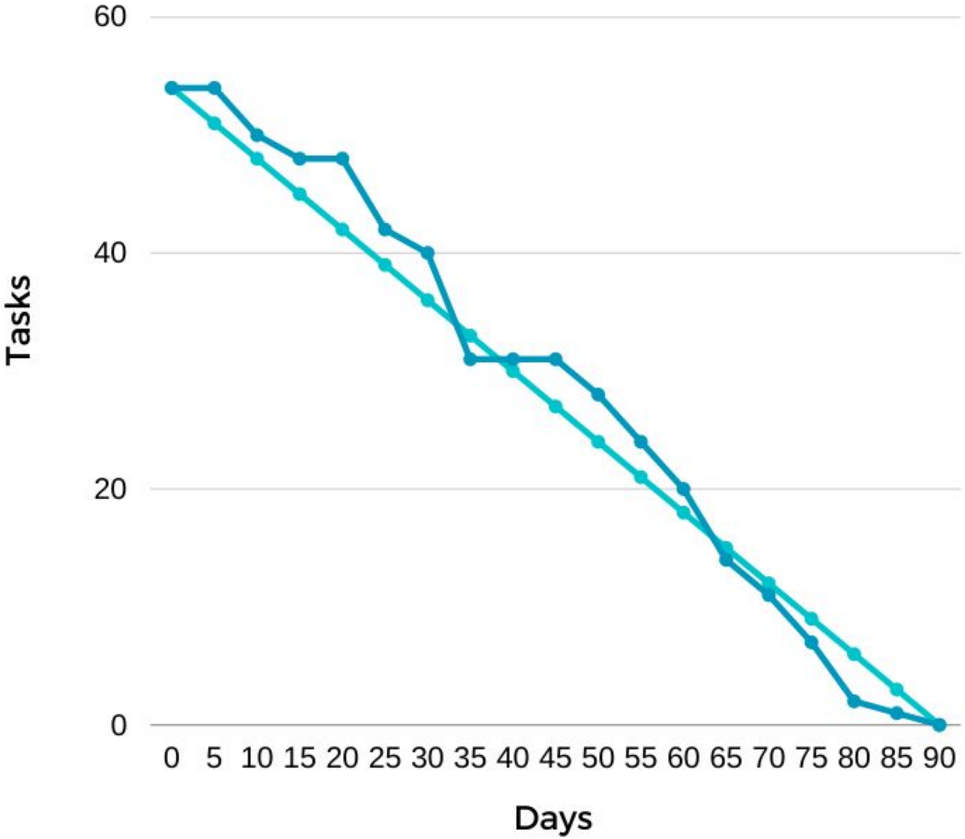
Design Decisions

- Decided to use Django and bootstrap due to having team members who were knowledgeable using them.
- Decided to Use the School‘s pre-written colors and fonts to ensure that the application fit in at USU.

SCRUM Practices

- Standup meetings were held once a week
- Sprint planning meetings were used to help keeps the sprints tasks organized
- Sprint Retrospective reports were held at the end of each sprint to ensure work completion and correctness.

## Final Burndown Chart



# Requirement: Create an Account

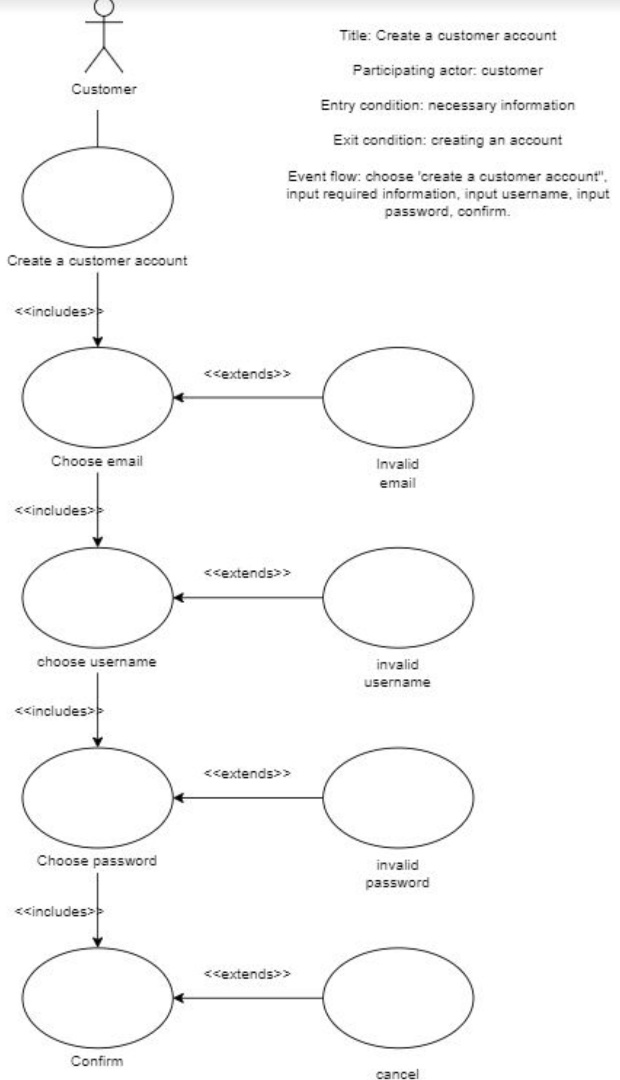
This requirement is essential for implementation. It is a “Must” according to the MOSCOW method. The audience for this requirement is for every type of user, as each user can create a basic account. It is a functional requirement for anyone to be able to create an account. These accounts are standard user accounts without any elevated permissions. Only supervisors can change a user’s permissions.

The description of this requirement according to the FURPS method is:

- **Functionality** - Anybody should be able to create a standard user account using a username, email, and password.
- **Usability** - (Please see next slides for a description of the usability)
- **Reliability** - This functionality should be available at all times and 100% accurate.
- **Performance** - This is a basic requirement that should perform very efficiently.
- **Supportability** - This requirement is available on all up to date browsers, both for mobile devices and desktop users. It is also an easy to maintain requirement.

# Create an Account

This requirement is located in the Requirements definition, labeled requirement 3.4. The UML to the right shows our design. Upon starting up the system, the user will see a screen that prompts them to log in. However, as the user has no account yet, they will instead select ‘create new account’. Upon selecting this option, the user will then be prompted to enter a email, username, and password. Once the user has entered all of these fields, they can then select ‘register’ and will be directed to the User dashboard, where they can then continue onto other functionalities and options, such as reserving a parking space.



Title: Create a customer account

Participating actor: customer

Entry condition: necessary information

Exit condition: creating an account

Event flow: choose 'create a customer account", input required information, input username, input password, confirm.

# Create an Account

Implementing this functionality required the following tasks and was completed in Sprint 1:

**Create the registration view function:**

This task was completed by Derek Ward

**Create the registration html template:**

This task was completed by Makalee Beelek

**Test the registration functionality:**

This task was completed by Anne Jetton

# Requirement: A Customer Can Make a Reservation

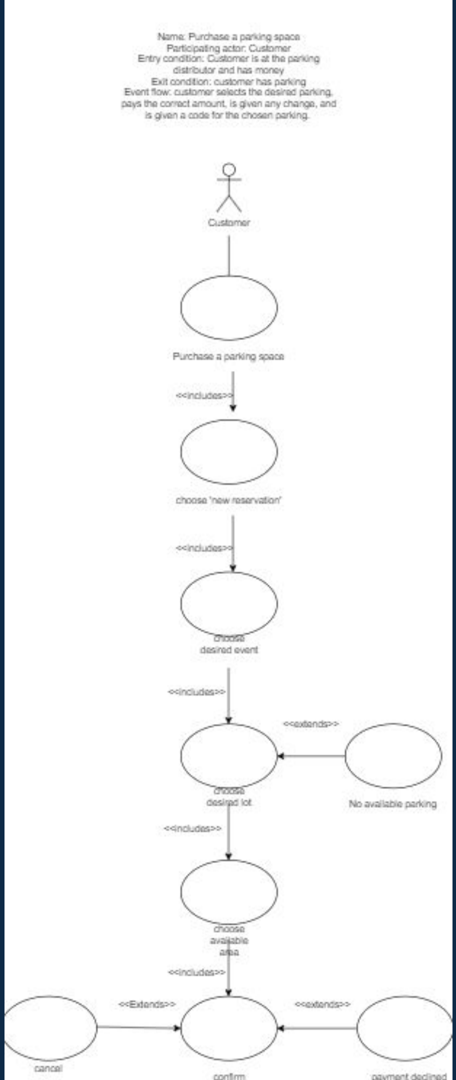
This requirement is essential for implementation. It is a “must” according to the MOSCOW method. The audience for this requirement is for every type of user, as each user can make reservations.  It is a functional requirement for users who will want to reserve parking spaces for events. To implement this requirement, we will need a user to be logged in, a supervisor will have had to create an event, and a manager to have posted a parking spot for reservation in the specific event.

The description of this requirement according to the FURPS method is:

- **Functionality** - Anybody should be able to create a standard user account using a username, email, and password.
- **Usability** - (Please see next slides for a description of the usability)
- **Reliability** - This functionality should be available at all times and 100% accurate.
- **Performance** - This is a basic requirement that should perform very efficiently.
- **Supportability** - This requirement is available on all up to date browsers, both for mobile devices and desktop users. It is also an easy to maintain requirement.

# A Customer Can Make a Reservation

This requirement is located in the Requirements definition, labeled requirement 3.1. The UML to the right shows our design. Once the user has logged into their account, they will find themselves on the dashboard. From there, the user can select the ‘new reservation’ button.  From there the user will be directed to the event listings, where they can select the event that they wish to purchase parking for. Simply clicking on the desired event will then direct the user to the available parking lots registered for that event. Choosing a lot will direct the user to the available areas in that lot, which is where a user can choose what kind of parking they will need, such as handicapped parking. A pop up will appear asking you to confirm all of the information, and pay for your space. Once the reservation has been confirmed, the user can then see the new reservation on there dashboard.



Name: Purchase a parking space
Participating actor: Customer
Entry condition: Customer is at the parking distributor and has money
Exit condition: customer has parking
Event flow: customer selects the desired parking, pays the correct amount, is given any change, and is given a code for the chosen parking.

# A Customer Can Make a Reservation

Implementing this functionality required the following tasks and was completed in Sprint 3:

**Create the reservation view function:**

This task was completed by Anne Jetton

**Create the reservation html template:**

This task was completed by Kate Sargent

**Test the reservation functionality:**

This task was completed by Makalee Beelek

# Requirement: A Customer Can Cancel A Reservation

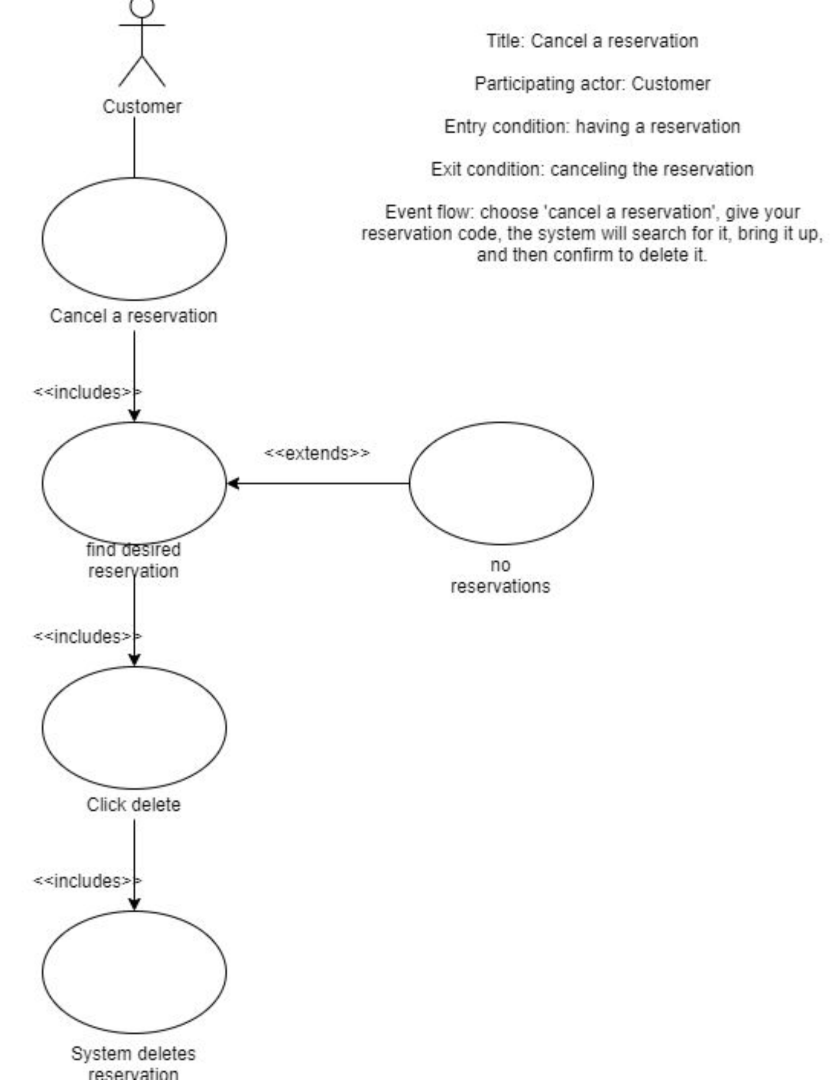
This requirement is essential for implementation. It is a “Must” according to the MOSCOW method. The audience for this requirement is for every type of user, as each user can reserve a reservation and cancel it. It is a functional requirement for anyone to be able to cancel a reservation and receive a refund. Any of the four user types can cancel a reservation. A reservation cannot be cancelled if it is within 24 hours of the event.

The description of this requirement according to the FURPS method is:

- **Functionality** - Anyone with any account type who has previously made a reservation should be able to cancel the reservation
- **Usability** - (Please see next slides for a description of the usability)
- **Reliability** - This functionality should be available at all times and 100% accurate.
- **Performance** - This is a basic requirement that should perform very efficiently.
- **Supportability** - This requirement is available on all up to date browsers, both for mobile devices and desktop users. It is also an easy to maintain requirement.

# A Customer Can Cancel A Reservation

This requirement is located in Requirements Definition document as requirement number 3.2. The UML to the left shows our design. Before deleting a reservation, there must be an event, lot, and subsequent areas posted by a supervisor or manager. The user can then create a reservation (detailed in prior slides). To cancel this reservation, a user must be logged in and viewing their dashboard. They will then need to locate the reservation in their “Your Reservations” list, then click on the “X Cancel” button to remove the reservation. The reservation will disappear and a refund will be added to their balance.



Title: Cancel a reservation

Participating actor: Customer

Entry condition: having a reservation

Exit condition: canceling the reservation

Event flow: choose 'cancel a reservation', give your reservation code, the system will search for it, bring it up, and then confirm to delete it.

# A Customer Can Cancel A Reservation

Implementing this functionality required the following tasks and was completed in Sprint 3:

**Create the reservation view function:**

This task was completed by Derek Ward

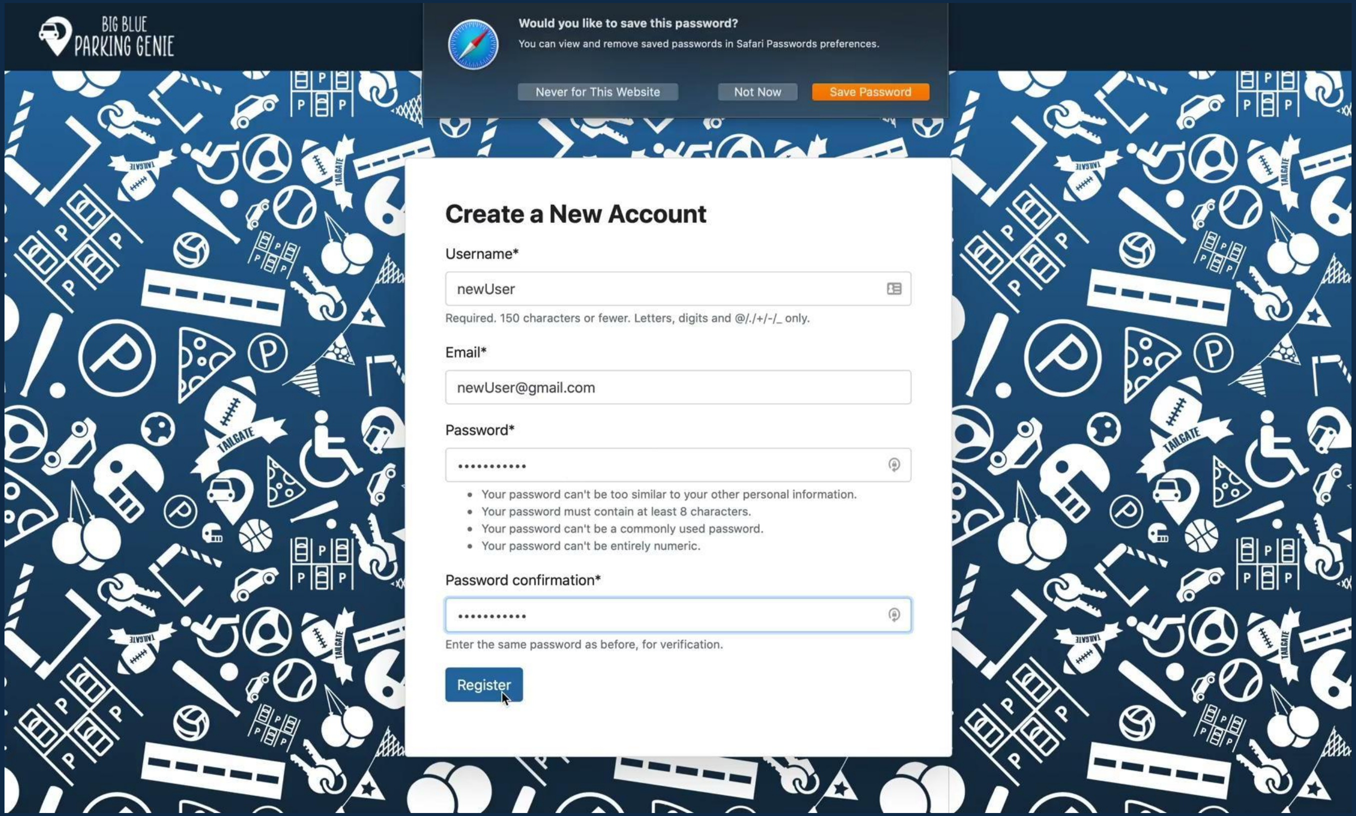
**Create the reservation html template:**

This task was completed by Anne Jetton

**Test the reservation functionality:**

This task was completed by Kate Sargent

# Create an Account (Recording)



Would you like to save this password?

You can view and remove saved passwords in Safari Passwords preferences.

## Create a New Account

Username*

newUser

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

newUser@gmail.com

Password*

••••••••••••

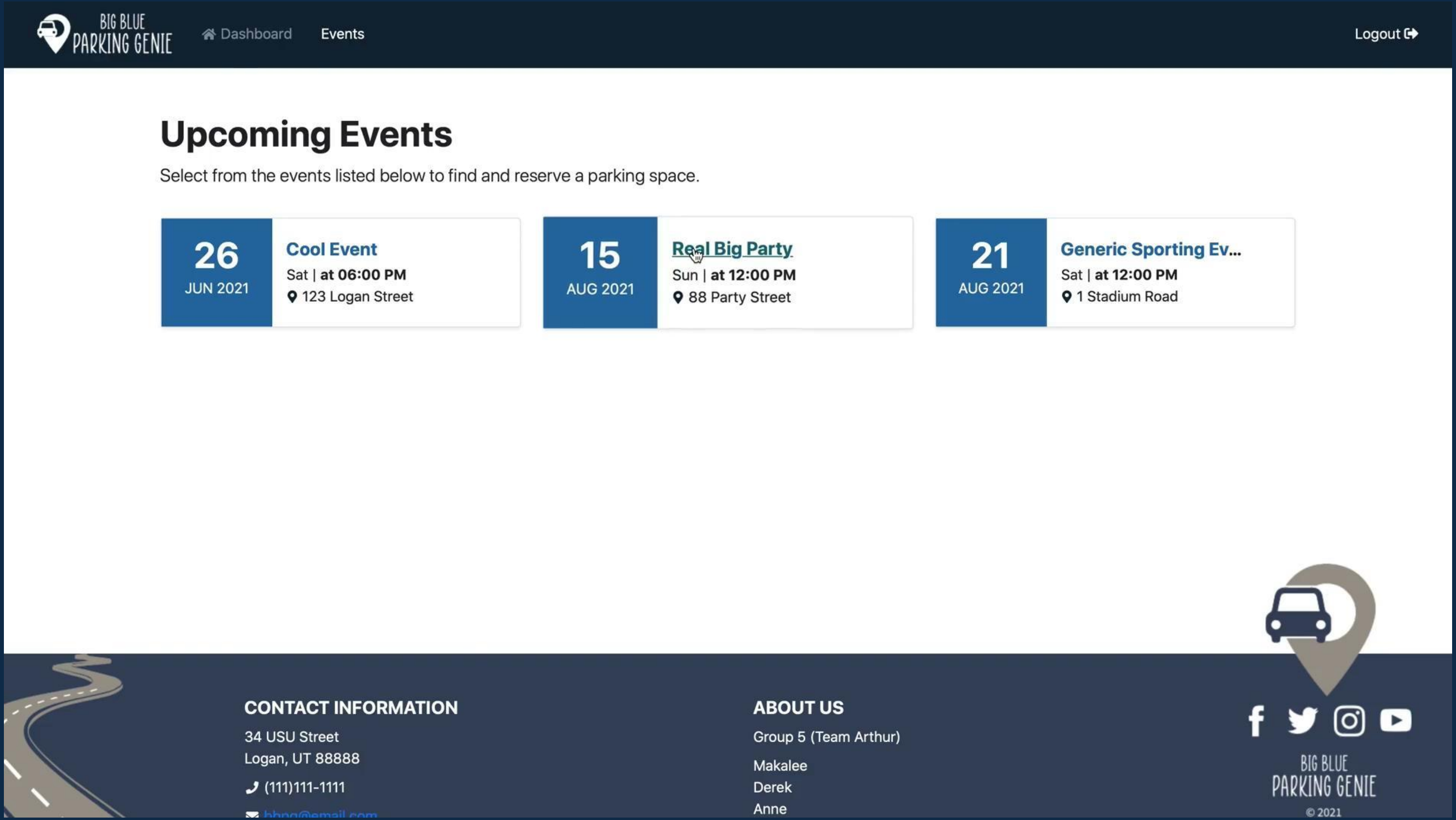
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

••••••••••••

Enter the same password as before, for verification.

# Reserve a Space (Recording)



## Upcoming Events

Select from the events listed below to find and reserve a parking space.

26 JUN 2021 — **Cool Event**
Sat | **at 06:00 PM**
123 Logan Street

15 AUG 2021 — **Real Big Party**
Sun | **at 12:00 PM**
88 Party Street

21 AUG 2021 — **Generic Sporting Ev...**
Sat | **at 12:00 PM**
1 Stadium Road

**CONTACT INFORMATION**

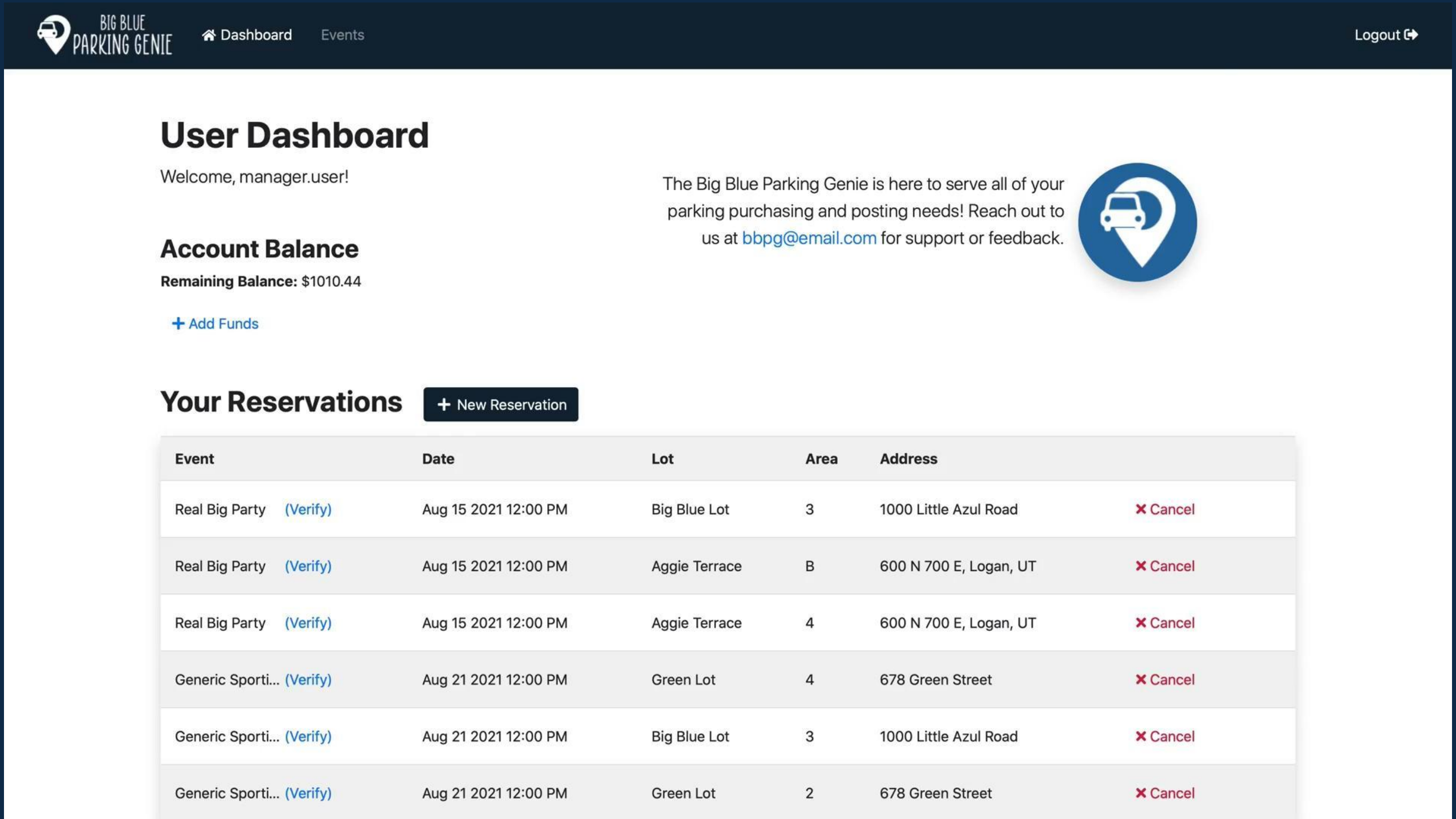
34 USU Street
Logan, UT 88888
☎ (111)111-1111
✉ bbpg@email.com

**ABOUT US**

Group 5 (Team Arthur)

Makalee
Derek
Anne

# Delete a Reservation (Recording)



## User Dashboard

Welcome, manager.user!

The Big Blue Parking Genie is here to serve all of your parking purchasing and posting needs! Reach out to us at bbpg@email.com for support or feedback.

### Account Balance

**Remaining Balance:** $1010.44

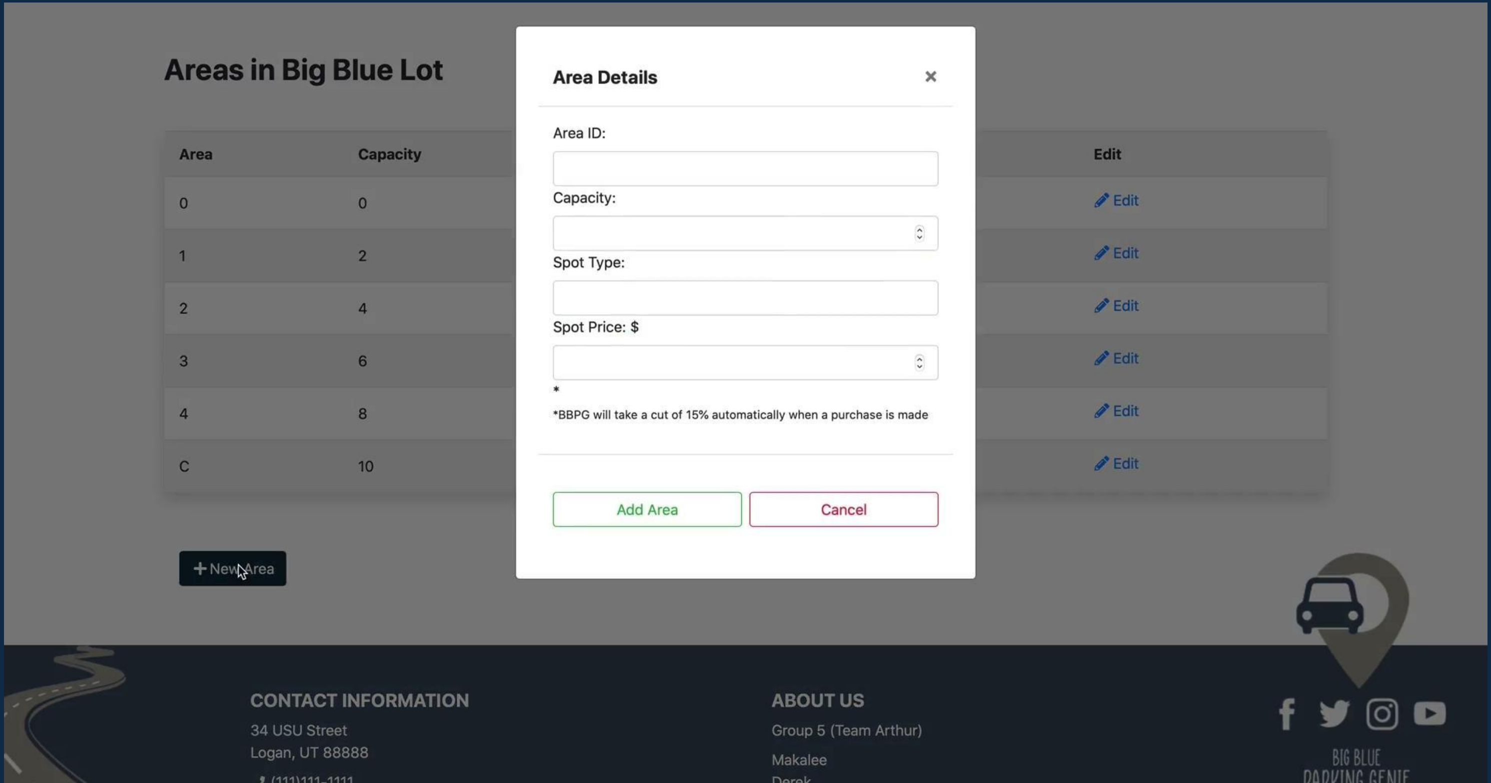
+ Add Funds

### Your Reservations

+ New Reservation

| Event | | Date | Lot | Area | Address | |
|---|---|---|---|---|---|---|
| Real Big Party | (Verify) | Aug 15 2021 12:00 PM | Big Blue Lot | 3 | 1000 Little Azul Road | ✕ Cancel |
| Real Big Party | (Verify) | Aug 15 2021 12:00 PM | Aggie Terrace | B | 600 N 700 E, Logan, UT | ✕ Cancel |
| Real Big Party | (Verify) | Aug 15 2021 12:00 PM | Aggie Terrace | 4 | 600 N 700 E, Logan, UT | ✕ Cancel |
| Generic Sporti... | (Verify) | Aug 21 2021 12:00 PM | Green Lot | 4 | 678 Green Street | ✕ Cancel |
| Generic Sporti... | (Verify) | Aug 21 2021 12:00 PM | Big Blue Lot | 3 | 1000 Little Azul Road | ✕ Cancel |
| Generic Sporti... | (Verify) | Aug 21 2021 12:00 PM | Green Lot | 2 | 678 Green Street | ✕ Cancel |

# Post a New Space (Recording)



## Areas in Big Blue Lot

| Area | Capacity | Edit |
|---|---|---|
| 0 | 0 | Edit |
| 1 | 2 | Edit |
| 2 | 4 | Edit |
| 3 | 6 | Edit |
| 4 | 8 | Edit |
| C | 10 | Edit |

### Area Details

Area ID:

Capacity:

Spot Type:

Spot Price: $

*

*BBPG will take a cut of 15% automatically when a purchase is made

CONTACT INFORMATION

34 USU Street
Logan, UT 88888
(111)111-1111

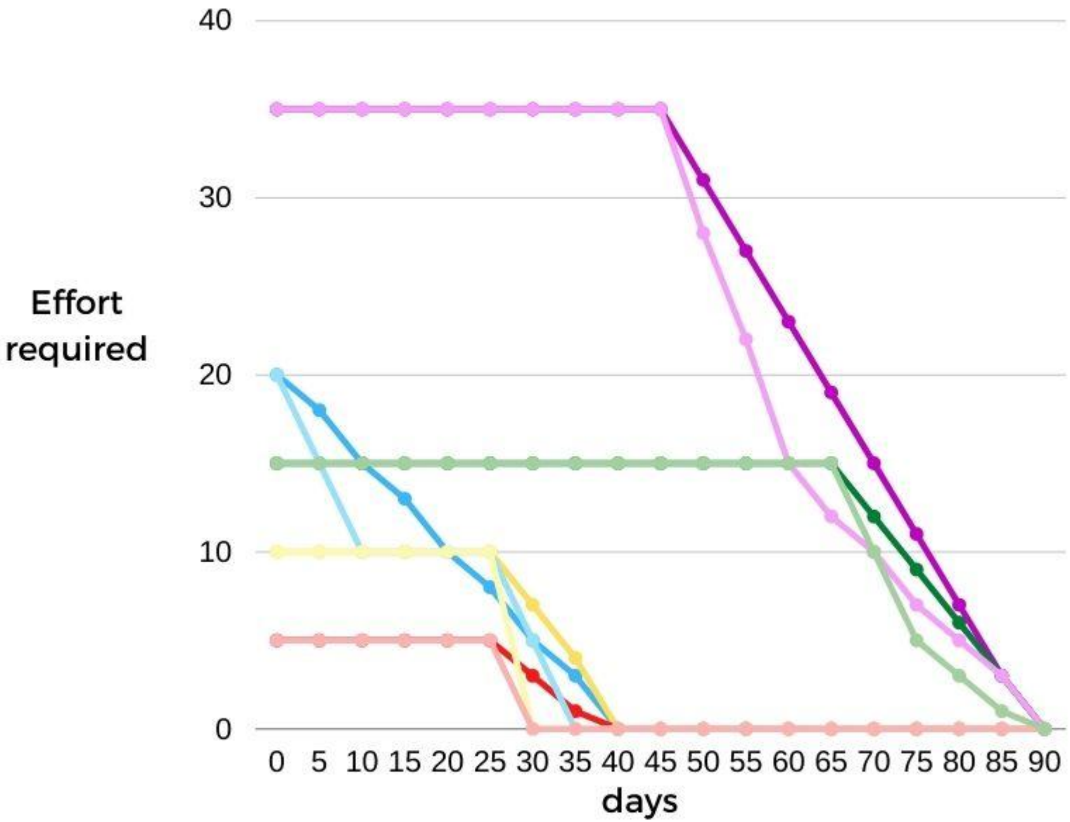
ABOUT US

Group 5 (Team Arthur)

Makalee
Derek

# Line Graph

Blue: Requirements, Yellow: Hi Level Design, Red: Low Level Design, Purple: Development, Green: Testing



# Resources

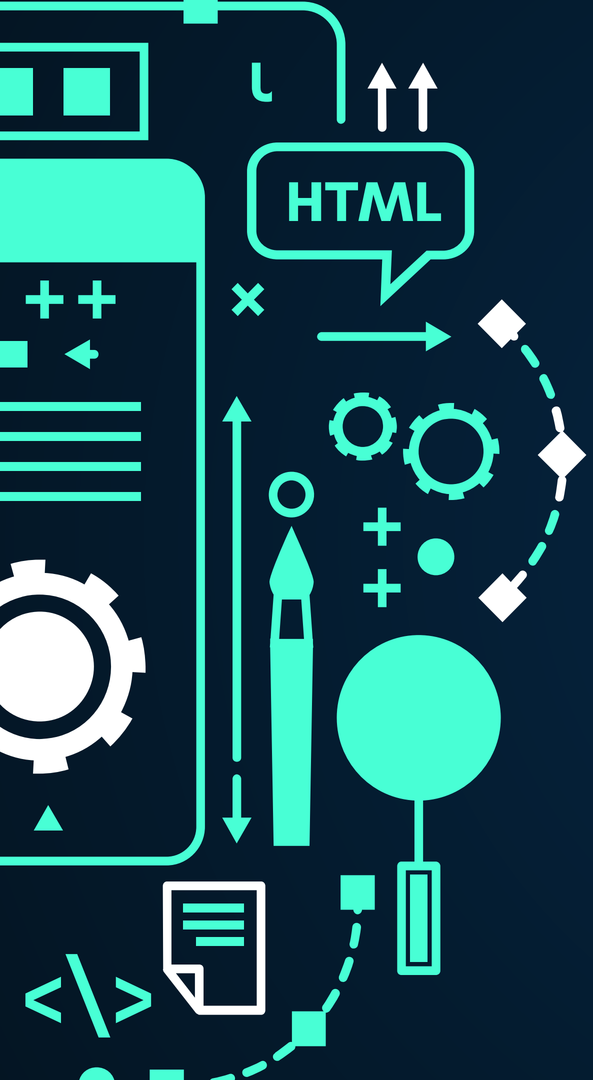
Django Software Foundation. (2005 - 2021). Django Documentation. Location of publisher not given.
https://docs.djangoproject.com/en/3.1/

Stephens, R. (2015). Beginning Software Engineering. Birmingham, England: Wrox Press Ltd.

W3Schools. (1999-2021). Bootstrap 4 All CSS Classes. Location of publisher not given.
https://www.w3schools.com/bootstrap4/bootstrap_ref_all_classes.asp

Mozilla. (2005-2021). JavaScript Reference, Location of publisher not given.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

Mozilla. (2005 - 2021). CSS: Cascading Style Sheets. Location of publisher not given.
https://developer.mozilla.org/en-US/docs/Web/CSS



# Questions?!