User Stories

Actors

Spectator - Guest of web application who either hasn't logged into account or doesn't own an account. They have restricted access to features offered by application.

Regular User - Client who have an account and are currently logged into account. They have ability to utilize features offered by application.

Administrator - Regular user who has special privileges and assigned special roles. They have authority to act upon a troubled scenario.

Story #1

Name: Account Registration
Actors: Spectator
Triggers/Preconditions: Using features, not logged in and has no existing account
Actions/Postconditions: Display sign up form, create new account
Acceptance Tests: *Username*, presence: true, length: {minimum: 5}, restrictions: {a-z, A-Z, 0-9, -, and _}; *firstname,* presence: true; last*name,* presence: true; *email*, presence: true, restrictions: {a-z, A-Z, 0-9, ., @, and _}, format: {end with @___.__}; *password*, presence: true, length: {minimum: 6}, restrictions: {one each of a-z, A-Z and 0-9}
assert *username* doesn't exist in database
Required Behavior: Display error message for acceptance test failure; new account created for acceptance test success
Relevance: Increase members
Iteration: 1

Account Registration

Given a spectator is browsing web application, when they are trying to access features while currently aren't logged in and has no existing account, then a sign up form appears which permits ability to create a new account.

Story #2

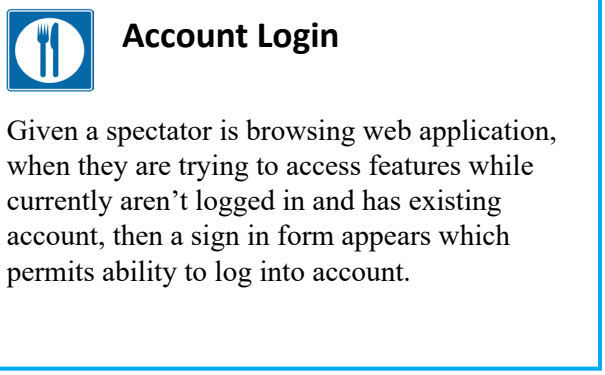
Name: Registration Verification
Actors: Spectator
Triggers/Preconditions: Successful Registration
Actions/Postconditions: Send verification e-mail
Acceptance Tests: -
Required Behavior: -
Relevance: Increase members
Iteration: 1

Registration Verification

Given a spectator is browsing web application, when they successfully register a new account with application, then a verification e-mail is sent to spectator.

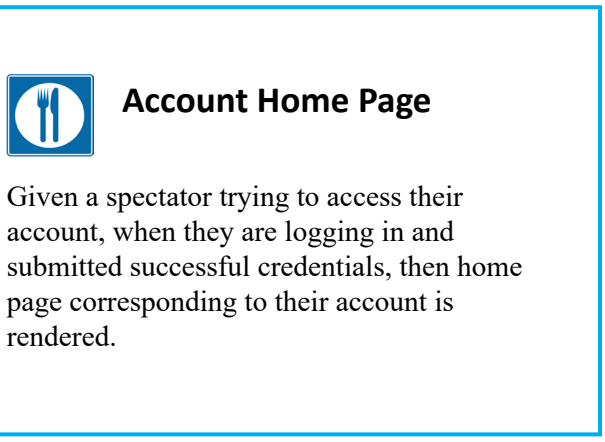
Story #3

Name: Account Login
Actors: Spectator
Triggers/Preconditions: Using features, not logged in and has existing account
Actions/Postconditions: Display sign in form, log into account
Acceptance Tests: *Username*, presence: true; *password*, presence: true
assert *e-mail* exists in database
assert *e-mail* and *password* match in database
Required Behavior: Display error message for acceptance test failure; logged into account for acceptance test success
Relevance: Increase productivity
Iteration: 1



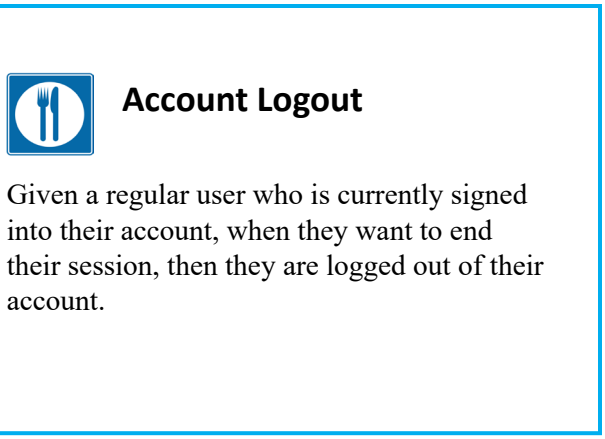
Story #4

Name: Account Home Page
Actors: Spectator, regular user
Triggers/Preconditions: Log into account, successful credentials submitted
Actions/Postconditions: Render home page
Acceptance Tests: -
Required Behavior: -
Relevance: Convenience
Iteration: 1



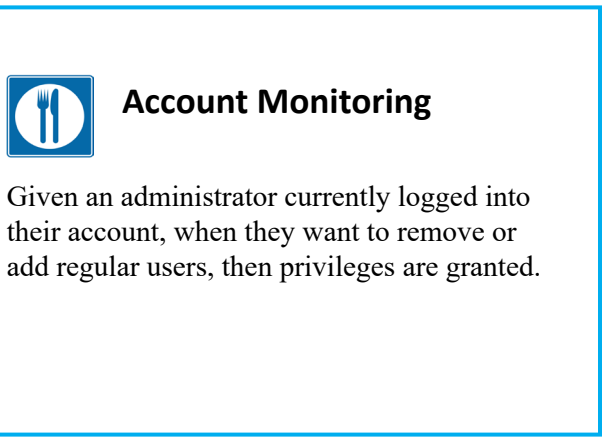
Story #5

Name: Account Logout
Actors: Regular user
Triggers/Preconditions: End session
Actions/Postconditions: Logged out
Acceptance Tests: -
Required Behavior: -
Relevance: Security
Iteration: 1



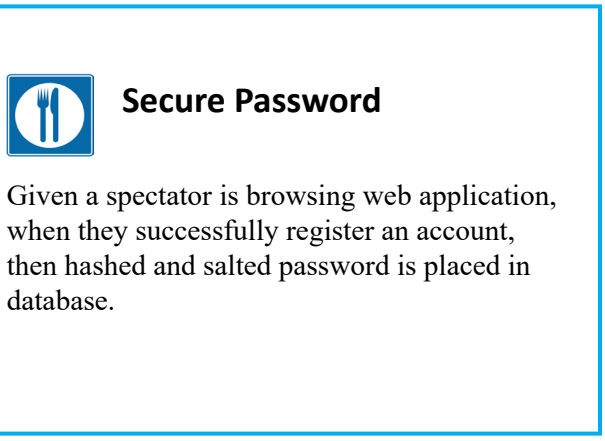
Story #6

Name: Account Monitoring
Actors: Administrator
Triggers/Preconditions: Username "masteruser" as administrator logged into account
Actions/Postconditions: Render user list, display remove regular users form and add users form, update regular users
Acceptance Tests: *Username*, presence: true, length: {minimum: 5}, restrictions: {a-z, A-Z, 0-9, -, and _}; *firstname,* presence: true; last*name,* presence: true; *email*, presence: true, restrictions: {a-z, A-Z, 0-9, ., @, and _}, format: {end with @___.__}; *password*, presence: true, length: {minimum: 6}, restrictions: {one each of a-z, A-Z and 0-9}
assert *username* doesn't exist in database
Required Behavior: Display error message for acceptance test failure; inform administrator of action change for acceptance test success
Relevance: Convenience, security
Iteration: 1 (continuation in iteration 2)



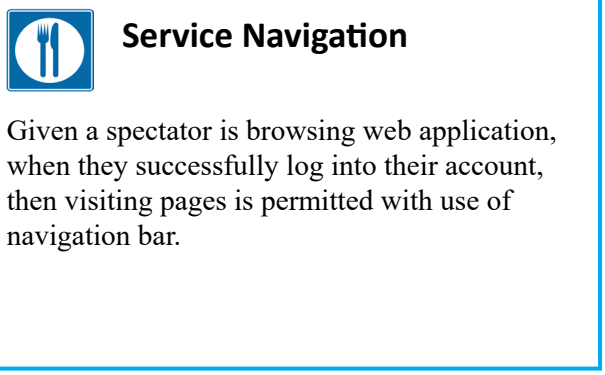
Story #7

Name: Secure Password
Actors: Spectator
Triggers/Preconditions: Account registration, successful credentials submitted
Actions/Postconditions: Create new account
Acceptance Tests: -
Required Behavior: -
Relevance: Security
Iteration: 1



Story #8

Name: Service Navigation
Actors: Spectator, regular user
Triggers/Preconditions: Visit pages
Actions/Postconditions: Rendering desired pages
Acceptance Tests: *Username [session],* presence: true
assert *username [session]* matches *username*
Required Behavior: Redirect to login and registration page upon acceptance test failure; redirect to desired page with navigation link click upon acceptance test pass
Relevance: Convenience
Iteration: 2

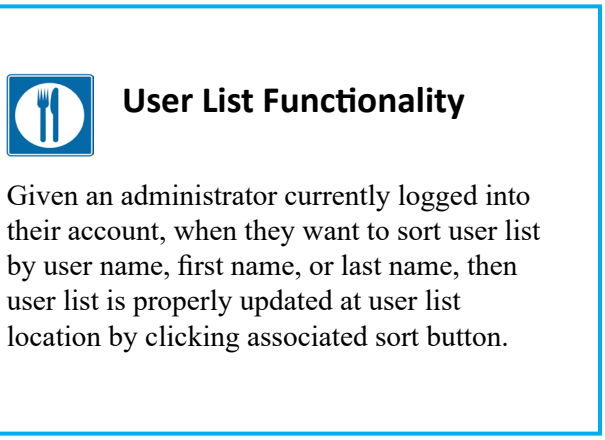


Story #9

Name: Restaurant for Dining
Actors: Regular user
Triggers/Preconditions: Log into account, successful credentials submitted, visit restaurant page
Actions/Postconditions: Display weekly restaurant used for dining
Acceptance Tests: -
Required Behavior: Output restaurant name if restaurant image failed to render properly
Relevance: Convenience
Iteration: 2 (continuation in iteration 3)

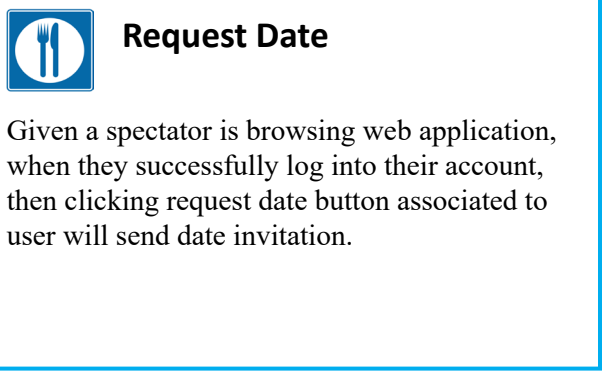
Story #10 (Continuation of Story #6)

Name: User List Functionality
Actors: Administrator
Triggers/Preconditions: <Story #6>, click sort by user name, first name, or last name button
Actions/Postconditions: <Story #6>, sort user list by user name, first name, or last name
Acceptance Tests: -
Required Behavior: <Story #6>, properly update user list at user list location
Relevance: <Story #6>
Iteration: 2



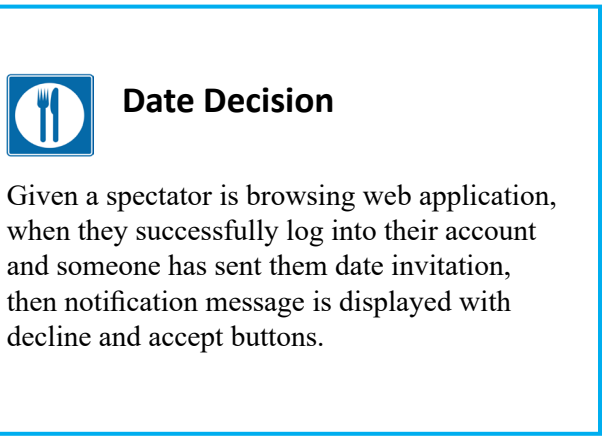
Story #11

Name: Request Date
Actors: Regular user
Triggers/Preconditions: Log into account, successful credentials submitted, request date button associated to user is clicked
Actions/Postconditions: Requested date
Acceptance Tests: -
Required Behavior: Update user in database who received date invitation
Relevance: Convenience,
Iteration: 2



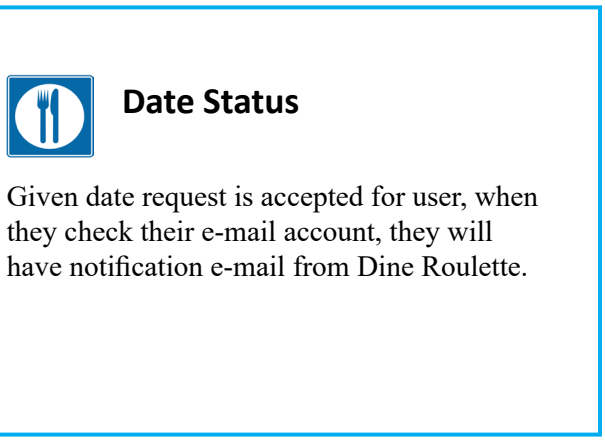
Story #12

Name: Date Decision
Actors: Regular user
Triggers/Preconditions: Log into account, successful credentials submitted, user receives date invitation
Actions/Postconditions: Display notification message with decline and accept buttons
Acceptance Tests: -
Required Behavior: Erase invitation by clicking decline button and continue invitation process by clicking accept button
Relevance: Convenience



Story #13

Name: Date Status
Actors: Regular user
Triggers/Preconditions: Log into account, successful credentials submitted, user accepts invitation
Actions/Postconditions: Notification e-mail is sent to user who requested date and their request is accepted
Acceptance Tests: -
Required Behavior: -
Relevance: Convenience
Iteration: 2 (continuation in iteration 3)



Story #14 (Continuation of Story #9)

Name: Randomized Restaurant
Actors: Regular user
Triggers/Preconditions: <Story #6>
Actions/Postconditions: <Story #6>, restaurant of week for dining is randomized on weekly basis
Acceptance Tests: -
Required Behavior: <Story #6>, changed in relation to timestamp
Relevance: Convenience
Iteration: 3

Randomized Restaurant

Given a spectator is browsing web application, when they successfully log into their account, then visiting restaurant page with navigation bar will display weekly randomized restaurant used for dining.

Story #15 (Continuation of Story #13)

Name: Notification E-mail
Actors: Regular user
Triggers/Preconditions: <Story #7>, continue with user invitation process
Actions/Postconditions: Notification e-mail is sent to participants upon date cancellation or completed Paypal payment
Acceptance Tests: -
Required Behavior: -
Relevance: Convenience
Iteration: 3

Notification E-mail

Given date request is cancelled or Paypal payment is completed, when they check their e-mail account, they will have notification e-mail from Dine Roulette.

Story #16

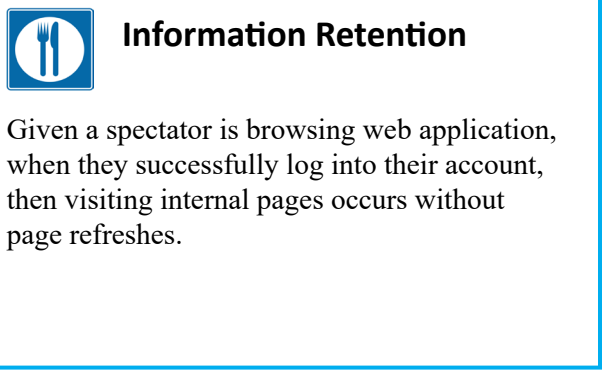
Name: Secondary Information
Actors: Regular user
Triggers/Preconditions: Clicking user
Actions/Postconditions: User's secondary information is rendered
Acceptance Tests: -
Required Behavior: -
Relevance: Convenience
Iteration: 3

Secondary Information

Given a spectator is browsing web application, when they successfully log into their account, then clicking a user will render following user's secondary information.

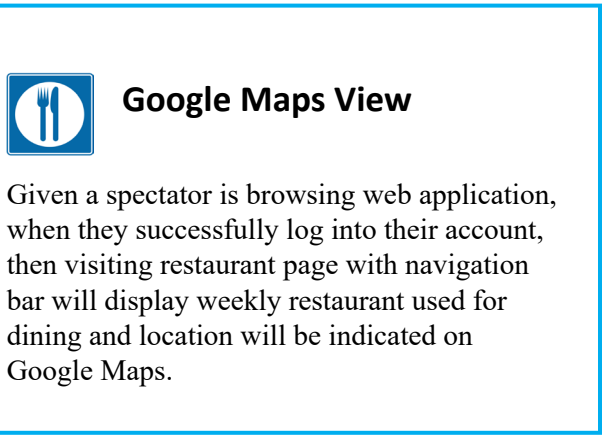
Story #17

Name: Information Retention
Actors: Regular user
Triggers/Preconditions: Switching between pages, regular user logged into service
Actions/Postconditions: Execute AJAX calls
Acceptance Tests: -
Required Behavior: Switch between internal pages without page refresh
Relevance: Convenience
Iteration: 3



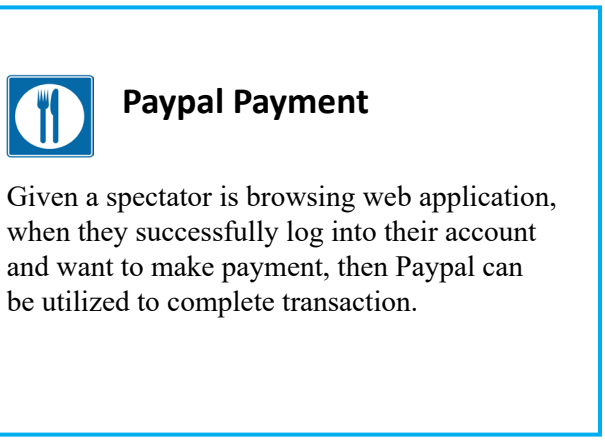
Story #18

Name: Google Maps View
Actors: Regular user
Triggers/Preconditions: Log into account, successful credentials submitted, visit restaurant page
Actions/Postconditions: Display Google Maps
Acceptance Tests: -
Required Behavior: -
Relevance: Convenience
Iteration: 3



Story #19

Name: Paypal Payment
Actors: Regular user
Triggers/Preconditions: User makes payment with Paypal
Actions/Postconditions: Paypal is utilized with assistance from Dine Roulette application
Acceptance Tests: -
Required Behavior: -
Relevance: Convenience
Iteration: 3



Velocity

Convention

User stories can be used to measure progress. This is done by assigning integer values called Story Points to user stories in accordance to their level of difficulty. By accumulating total story points we are able to dictate rate of delivery called Velocity.

Story Points - 1 means straight forward
2 means intermediate
3 means complex

Velocity - Average number of story points per iteration.

Velocity of Iteration #1

Stories Implemented

#1 - Account Registration - Story Points: 2
#2 - Registration Verification - Story Points: 1
#3 - Account Login - Story Points: 1
#4 - Account Home Page - Story Points: 1
#5 - Account Logout - Story Points: 1
#6 - Account Monitoring - Story Points: 2
#7 - Secure Password - Story Points: 1

DineRoulette

Velocity of iteration #1 is 9.

Velocity of Iteration #2

Stories Implemented

#8 - Service Navigation - Story Points: 1
#9 - Restaurant for Dining - Story Points: 1
#10 - User List Functionality - Story Points: 2
#11 - Request Date - Story Points: 2
#12 - Date Decision - Story Points: 2
#13 - Date Status - Story Points: 2

DineRoulette

Velocity of iteration #2 is 10.

Velocity of Iteration #3

Stories Implemented

#14 - Randomized Restaurant - Story Points: 2
#15 - Notification E-mail - Story Points: 1
#16 - Secondary Information - Story Points: 1
#17 - Information Retention - Story Points: 2
#18 - Google Maps View - Story Points: 2
#19 - Paypal Payment - Story Points: 2

DineRoulette

Velocity of iteration #3 is 10.