Use Case # 1: Search user by region (Major Use Case)

GENERAL CHARACTERISTICS	
Author	Gil Turner, Irtiza Delwar, Sung Kim
Last Update:	9/19/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization
Primary Actor	User of the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Does the searching based on the username Riot Games - Show statistics for the username searched. Owner of username - WIll have their statistics and information displayed on the website.
Preconditions	User has navigated to the site and the site is currently operational.
Success Post Condition	Information and data about the desired user has been successfully displayed.
Failed Post Condition	Username is invalid for the selected region Failure to fetch the data needed

MAIN SUCCESS SCENARIO (or basic flow)

Step	**Action -** description in words of each step in success scenario
1	User goes to the website application
2	User types in username in search box
3	The user selects which region they want to search the username for and then clicks search.
4	The website displays information based on the username and region after retrieving said information from the Riot Games API

EXTENSIONS or Alternate Flows

Step	Branching Action
3b	Invalid username for the region
	1. API call says that the username is invalid or not found. 2. Website returns an error message saying that it cannot find the desired username for the selected region. 3. The user checks and re-enters the username and region and clicks search. 4. Repeat steps 2-3 until the user has correctly entered the username and region.
4b	API Call Fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	Requirement

1	The region search feature should be a drop down menu so the user doesn’t have to manually enter in the name of the region
2	The website should save previous search memory. Showing previous user names that were searched when the search box is clicked.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
1	The region search must be able to search with every region that the Riot Games API allows.

FREQUENCY OF OCCURRENCE: HIGH

OTHER ISSUES

Issue Num	**Issue**
1	Remember the last username that has been searched
2	The error message if no user can be found would be displayed on the search page instead of going to a new page.

Use Case # 2: Specific Champion Mastery

GENERAL CHARACTERISTICS	
Author	Gil Turner, Irtiza Delwar, Sung Kim
Last Update:	9/19/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization
Primary Actor	User, using the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Sees the champion mastery Riot Games - Show statistics for the username searched. Owner of username - WIll have their statistics and information displayed on the website.
Preconditions	The user has entered in a successful search
Success Post Condition	The website will display the correct champion mastery level for the desired champion of the username inputted.
Failed Post Condition	API does not find data for the specific data for the champion mastery.

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario

1	The user has done a successful username search and will then navigate to the champion mastery section.
2	In the champion mastery section, the user will select the name of the champion they want to search the username's mastery for.
3	The user then clicks the search mastery button.
4	The backend will perform an API call to retrieve the desired champion mastery for the user.
5	The front end will see the API call results and post the information onto the website.
6	The user then sees the mastery level for the champion they have selected for the desired champion.
7	The user can then repeat steps 2-6 with as many champions as they desire.

EXTENSIONS or Alternate Flows

Step	Branching Action
6a	Incorrect champion is selected
	6a1. The user notices that the champion they searched for was incorrect 6a2. The user then repeats steps 2-6 until they correctly select and receive information about the champion they wanted to know more about.
2a	The user wants all the champion masteries
	2a1.The user would then follow the steps as described in Use Case 3.
5b	API Call Fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	The user can only chose 1 specific champion at a time
2	The champion list should be a menu with all the champions listed so the user doesn’t have to write in the name of the champion.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
0	N/A

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES

Issue Num	**Issue**
1	Under the champion list menu, include a small picture of the champion
2	When returning the champion mastery level, also return the mastery score of the champion if possible.

Use Case # 3: Entire Champion Mastery

GENERAL CHARACTERISTICS

Author	Gil Turner, Irtiza Delwar, Sung Kim
Last Update:	9/19/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization.
Primary Actor	User, using the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Sees the champion mastery for every champion for that username the user specified. Riot Games - Show statistics for the username that the user searched for. Owner of username - WIll have their statistics and information displayed on the website.
Preconditions	The user has entered in a successful search.
Success Post Condition	The website will display the correct champion mastery level for all the champions.
Failed Post Condition	The API call was unsuccessful or the information displayed is incorrect.

MAIN SUCCESS SCENARIO (or basic flow)

Step	**Action -** description in words of each step in success scenario
1	The user has done a successful username search and will then navigate to the champion mastery section.
2	In the champion mastery section, the user will select the button that says “all champions”.
3	The user then clicks the search mastery button.
4	The backend will perform an API call to retrieve all the champion mastery levels for the user.
5	The front end will see the API call results and post the information onto the website.
6	The information about the champion mastery levels will be displayed in a nice and clean looking format.
7	The user sees this information and then can decide to navigate to somewhere else on the site or leave.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
3a	The user didn’t click the “all champions” button nor did they select a specific champion.
	1. When the user hits the search button an error message will popup notifying them to select a champion or all champions. 2. The user then decides if they want to select all champions or they meant to click a specific champion. 3. If they meant to select all champions, the user will go back to step 2 and proceed from there. If they meant to select a specific champion, they will then follow Use Case 2.
5b	API Call Fails 1. Unable to access the API

	2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	The list of all the champions and their corresponding mastery level is easy to read for the user.
2	The list will be returned in alphabetical order.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
1	In order to select all champions, there must be a specific button in order to perform this operation so that the user doesn’t have to click every single champion.

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES

Issue Num	**Issue**
1	Possibly add more ways to sort the results such as highest to lowest champion mastery level.

2	Possibly include the champion mastery score for each champion.

Use Case # 4: Ranked Seasonal Stats

GENERAL CHARACTERISTICS	
Author	Gil Turner, Irtiza Delwar, Sung Kim
Last Update:	9/19/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization.
Primary Actor	User, using the website.
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Performs the search based on the username Riot Games - Show statistics for the username searched. Owner of username - WIll have their statistics and information displayed on the website.
Preconditions	The user has entered in a successful search.
Success Post Condition	The statistics for the current ranked season have been correctly displayed and the user can easily read them.
Failed Post Condition	The API call was unsuccessful or the information displayed is incorrect.

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario
1	The user has done a successful username search.

2	The user will navigate to the “Seasonal Ranked Stats” button.
3	The user then clicks the “Seasonal Ranked Stats” button.
4	The back end will perform the API call to get the seasonal ranked stats such as the wins/loses, average kills, deaths, and assists, number of double, triple, quadra, and penta kills, etc.
5	The front end will see the API call results and post the information onto the website.
6	The website will display the statistics in a clean and ordered manner so that the user can easily understand everything
7	The user sees this information and then can decide to navigate to somewhere else on the site or leave.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
5b	API Call Fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	The statistical information is displayed in an easy to read manner.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
1	The data displayed should only be from the current ranked season.

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES	
Issue Num	**Issue**
1	Possibly allow the user to select which rank season they want data from.
2	Possibly allow the user to select what specific data they want from the ranked season via a menu.
3	Possibly show the information in some graphical form because graphs are easier to read and understand..

Use Case # 5: User Champion Statistics

GENERAL CHARACTERISTICS	
Author	Irtiza Delwar, Gil Turner, Sung Kim
Last Update:	9/19/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization
Primary Actor	User, using website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Finds statistics based on champions played in ranked games. Riot Games - Show statistics from their game. Owner of username - WIll have their statistics and information displayed on the website.
Preconditions	User has entered a successful search.
Success Post Condition	Statistics are displayed based on champion played.
Failed Post Condition	The API call was unsuccessful or statistics displayed incorrectly or the account has no games played this season.

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario

1	The user has done a successful username search.
2	The user will navigate to the “Champion Statistics” button.
3	The user then clicks the “Champion Statistics” button.
4	The back end will perform the API call to get the champion stats such as the wins/loses, kills, deaths, and assists, number of double, triple, quadra, and penta kills, etc.
5	The front end will see the API call results and post the information onto the website.
6	The website will display the statistics in a clean and ordered manner so that the user can easily understand everything
7	The user sees this information and then can decide to navigate to somewhere else on the site or leave.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
3a	The user clicks the wrong button
	1. The user sees the information that is displayed is not what they wanted. 2. The user goes back to the page that is directly after the successful search. 3. The user goes back to step 2 in this use case.
5b	API Call Fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	The statistical information is displayed in an easy to read manner.
2	The Champions are sorted in order of games played.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
1	The data displayed should only be from the current ranked season.

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES

Issue Num	**Issue**
1	Allow the user to sort by each of the columns of data (KDA, wins, etc).
2	Possibly show tips and critique based on statistics.
3	Possible add color code / pictures to differentiate performance on champions.

Use Case # 6: Ranked Team Stats

GENERAL CHARACTERISTICS	
Author	Gil Turner, Irtiza Delwar, Sung Kim
Last Update:	9/24/2016
Scope	Web Page
Level	User level
Status	Incomplete, conceptualization
Primary Actor	User of website.
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Does the searching based on the username Riot Games - Show statistics for the username searched. Owner of username - WIll have their statistics and information displayed on the website.
Preconditions	The user has already completed a successful search.
Success Post Condition	Data and statistics about the rank team have been correctly displayed.
Failed Post Condition	The ranked team that was searched for was invalid. Unable to receive data from API

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario

1	There will then be a ranked teams search bar where the user can input the name of the ranked team. User types the name of the ranked team
2	The user clicks “search for ranked team” button
3	The program performs the necessary API calls to retrieve information about the ranked team.
4	The information is parsed out into “team members”, win percentage, and match history” on the website.
5	The user can then analysis the data shown.
6	If the user wants to perform another search, return to step 2 else the user can navigate to another section on the website.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
3a	Invalid ranked team
	4a1. The ranked team the user searches for doesn’t exist so the website would return an error message 4a2. The user then validates that the ranked team name they entered was valid. 4a3. The user would then re-enter the ranked team name. 4a4. Return to step 3.
4b	API Call Fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	Requirement
1	The ranked team search should return at least 3 forms of data (users on the team, rank of the team, number of points the team has)about the ranked team
2	The search must return the win percentage of the ranked team
3	Information should be displayed in an easy to read manner.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	Variation
1	Should include a menu or buttons that would allow the user to get additional or more detailed information.

FREQUENCY OF OCCURRENCE: LOW

OTHER ISSUES

Issue Num	Issue
1	Possibly add the win percentage of each team member for the ranked team.
2	Possibly return the champions the ranked team has played the most.
3	Could add the feature that does the Champion Composition Recommendation (Use Care #8) for the ranked team.
4	Add charts and graphs of the data returned so that the information is easier to understand.

Use Case # 7: Match History

GENERAL CHARACTERISTICS	
Author	Irtiza Delwar, Gil Turner, Sung Kim
Last Update:	9/24/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization
Primary Actor	User, using the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Looks up their own or another person’s past match history Riot Games - Provides statistics and data about previous matches for the account that was searched Owner of username - WIll have their statistics and information displayed on the website.
Preconditions	A search username (Use Case #1) has been successfully completed.
Success Post Condition	The webpage displays the match history of the account that was searched.
Failed Post Condition	The webpage displays an error message saying that the user could not be found. The webpage displays that no match history could be found.

MAIN SUCCESS SCENARIO (or basic flow)

Step	**Action -** description in words of each step in success scenario
1	User goes to the website application
2	User completes a search for the account they are looking up
3	The API retrieves the information requested
4	The website displays the information included in a match history.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
4b	API Call Fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.
5	The user wants to see older matches in the matches displayed
	5a. The user scrolls to the bottom of the match history which is initially 10. 5b. The user clicks on the “see more” button 5c. The process returns to step 3 and repeats as necessary

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	Match history should have the ability to display older data than the last 10 games shown if requested.
2	The match history should be color coded based on wins/losses

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	Variation
1	Ability to sort between ranked and unranked match history, which will each be sorted by date respectively

FREQUENCY OF OCCURRENCE: HIGH

OTHER ISSUES

Issue Num	Issue
1	Summary of match history with tips to improve
2	Images based on performance in the match

Use Case # 8: Champion Composition Suggestion

GENERAL CHARACTERISTICS	
Author	Gil Turner, Irtiza Delwar, Sung Kim
Last Update:	9/24/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization
Primary Actor	User using the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Finds statistics based on champions played by the user. Riot Games - Show statistics/data from their game Owner of username - WIll have their statistics and information displayed on the website.
Preconditions	User has entered a successful search
Success Post Condition	The website has displayed accurate recommendations based on the match history of the username that was searched.
Failed Post Condition	The API calls that are required fails, the username searched doesn't have enough games played, or the username searched has a too wide variety of champions played.

MAIN SUCCESS SCENARIO (or basic flow)

Step	**Action -** description in words of each step in success scenario
1	The user has done a successful username search.
2	The user will navigate to the champion composition tab on the website
3	There will then be two buttons to click, either the “All games” or the “Ranked games”. One of the buttons would then be clicked.
4	API call is made and recommendation on a champion composition based on the match history is displayed.
5	The user then selects which role they want the champion composition to be filtered by.
6	The data then displays the specific role champion composition commendation recommendation that the user selected
7	The user can then look and think about our composition recommendation and then either go somewhere else on the site or leave.

EXTENSIONS or Alternate Flows	
Step	**Branching Action**
4a	The “All games” button is chosen
	4a1. When performing the API calls, the backend will pull out all the match history with no additional parsing needed
4b	The “Ranked games” button is chosen
	4b1. When performing the API calls, the backend will pull the match history and parse the games that are ranked only. 4b2. The user will then be prompted to ask whether they want ranked solo queue or ranked teams for their champion composition search 4b3. Then, our program will then need to parse between ranked solo queue and ranked teams.

4c	API Call Fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	This use case should have it’s own separate tab on the website

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
1	Must support both all and ranked games for the champion composition.
2	Should include some statistics explaining why our algorithm chose this champion composition. Our algorithm that determines the champion composition will take into account the win rate when certain champions are on the same team in the user’s last 50 games. This will be coded in Python.

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES

Issue Num	**Issue**
1	The information should be displayed in an easy to read manner.

2	There should be a menu that allows the user to sort through the champion composition recommendation by role after performing the initial search.
3	Possibly include some graphs and charts to make the data look nicer.

Use Case # 9: Live Match Predictor

GENERAL CHARACTERISTICS	
Author	Irtiza Delwar, Gil Turner, Sung Kim
Last Update:	9/24/2016
Scope	Web Page
Level	User-level
Status	Incomplete, Conceptualization
Primary Actor	User, using the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Checks who is predicted to win the match Riot Games - Provides statistics that are analyzed to predict winner of the match Players in the match - There will 10 players in the match that will have their information and data used for the predictor.
Preconditions	The website can be in any state that shows the navbar
Success Post Condition	Displays statistics of the 10 players in the game Displays prediction of who is likely to win the match
Failed Post Condition	The API is unable to find a match for the given player

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario

1	User clicks on the Live Match Predictor button on the website
2	User arrives at the Live Match Predictor page and locates the search bar
3	User Enters username of account to look up
4	API returns information based on the username entered
5	Program processes information to predict who will win the match based on statistics returned.
6	Front end receives information to show on website
7	The website will display the statistics in a clean and ordered manner so that the user can easily understand everything
8	The user sees this information and then can decide to navigate to somewhere else on the site or leave.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
4a	API Call Fails
	1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	The statistical information is displayed in an easy to read manner.

2	Displays prediction of who will win the match

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
0	N/A

FREQUENCY OF OCCURRENCE: LOW

OTHER ISSUES

Issue Num	**Issue**
1	Show messages for each player in the game that make conclusions based on the data. i.e. “ This player is on a 10 game win streak! ”
2	Color code teams based on who is predicted to win

Use Case # 10: Champion Analytics

GENERAL CHARACTERISTICS	
Author	Irtiza Delwar, Sung Kim, Gil Turner
Last Update:	9/24/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization
Primary Actor	User, using the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Viewing statistics based on champion Riot Games - API provides statistics and data about each champion
Preconditions	N/A
Success Post Condition	Web page displays all of the champions, allowing the user to pick one.
Failed Post Condition	Web page displays an error message

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario
1	User clicks on the Champion Analytics button on the navbar

2	User selects a champion from the list of champions
3	API retrieves information on the champion
4	Web page displays champion statistics
5	After seeing the champion statistics, the user can either leave the website or go somewhere else on the website

EXTENSIONS or Alternate Flows

Step	**Branching Action**
4a	API call fails
	1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.
5a	More detailed information
	5a1. Below the basic champion statistics, there would be a button that when clicked, would give a more detailed information and statistics about the champion. 5a2. The user clicks that button, then our website would display this more detailed information about the champion based on what was gathered from the API call.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	The user should be able to get a more detailed breakdown of the champion statistics.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	Variation
0	N/A

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES

Issue Num	Issue
1	Should display the stats in a graphical view so it’s easier to read.
2	Possibly allow the user to search for more than champions at once.

Use Case # 11: Top 5 Lists

GENERAL CHARACTERISTICS	
Author	Gil Turner, Irtiza Delwar, Sung Kim
Last Update:	9/24/2016
Scope	Web Page
Level	User-level
Status	Incomplete, conceptualization.
Primary Actor	User using the website.
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Viewing statistics based on champion Riot Games - API provides statistics and data about each champion
Preconditions	The user is on the website.
Success Post Condition	The data about top 5 champions based on win percentage and lost percentage has been accurately displayed.
Failed Post Condition	Searching the API failed and the information was not displayed on the website.

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario
1	The user navigates to the “Top 5” section of the website.

2	Once the user finds the section on the website, the user clicks that tab.
3	The API gathers the information and data regarding each champions win percentage.
4	The win percentage of each champion is then organized in a list from highest win rate to lowest win rate.
5	On the website, the list would be displayed with the champion and its corresponding win percentage based on the section (ex. Highest win rate section, lowest win rate section, etc.)
6	The user can then analysis the data and decide whether to stay on the website or leave the website.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
3b	API Call fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.
4a	Multiple Sections based on positions
	4a1: There should be multiple sections based on the 5 positions (Top, Jungle, Mid, ADC, Support) 4a2: The program should then perform API calls for each champion to see what the primary role is for the champion 4a3: With each champion now with a role, we then will be able to separate them to position list that is sorted based on win percentage.

SPECIAL REQUIREMENTS

Req	**Requirement**

Num	
1	System should include overall top 5 and bottom 5 win percentage champions as well as top and bottom 5 win percentage for each individual role.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	Variation
1	The win percentage pulled should only be from normal and ranked games. Doesn’t include speciality game modes or custom games.

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES

Issue Num	Issue
1	Possibly have the information displayed in a graphical view so that it is easier to read and understand.
2	Possibly have the history of the top and bottom 5 champions from a certain time frame (Ex: past month, past week, current day, etc.)
3	Could also have the website possibly display all the win percentage for each champion with a button after doing the initial search.

Use Case # 12: Free Champion Rotation

GENERAL CHARACTERISTICS	
Author	Irtiza Delwar, Gil Turner, Sung Kim
Last Update:	9/24/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization
Primary Actor	User, using the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Views the free champion Riot Games - Provides statistics and data about previous matches for the account that was searched
Preconditions	We are on any page of the web application
Success Post Condition	The web page displays the free champions
Failed Post Condition	The web pages displays an error message

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario
1	User goes to the website application

2	User clicks on free champion rotation button
3	API finds free champion rotation data
4	Web page displays free champion rotation

EXTENSIONS or Alternate Flows

Step	**Branching Action**
4a	API call fails
	1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	The picture of the champion should be displayed.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
0	N/A

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES

Issue Num	**Issue**
1	Possibly include how many days are left before the next free champion rotation.

Use Case # 13: Ranking Ladder

GENERAL CHARACTERISTICS	
Author	Irtiza Delwar, Gil Turner, Sung Kim
Last Update:	9/24/2016
Scope	Web Page
Level	User level
Status	Incomplete, Conceptualization
Primary Actor	User, using the website
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Looks up the top ranked players on the server Riot Games - Proves the top rankings on the server Users on the top of the ranking ladder - They will have their username and statistics displayed on the website.
Preconditions	User is on a web page on the website and can view the navbar
Success Post Condition	Web page displays the ranking ladder
Failed Post Condition	Web page displays an error message “Cannot connect to API”

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario
1	The user arrives at the web application

2	The user clicks on the ladder button
3	The program requests the ladder info from the api
4	Web page displays ladder information of the top 100 players in Challenger
5	The user then can navigate to somewhere else on the website or leave the website

EXTENSIONS or Alternate Flows

Step	**Branching Action**
4a	API call fails
	1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.
5a	The user wants all the challenger players
	5a1. The user would then look at the top of the list and click the “Show entire Challenger Ladder”. 5a2. The website would then show the entire challenger ladder list to the user. (This information was already gathered from the API call).

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	The ladder ranking is in a format that is easy to see and understand.
2	The search shows the top 100 players in challenger.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	Variation
0	N/A

FREQUENCY OF OCCURRENCE: LOW

OTHER ISSUES

Issue Num	Issue
1	Possibly include the entire challenger ladder.
2	Possibly include the master tier ladder

Use Case # 14: Ultimate Bravery

GENERAL CHARACTERISTICS	
Author	Sung Ho Kim, Gil Turner, Irtiza Delwar
Last Update:	9/24/2016
Scope	Web Page
Level	User Level
Status	Incomplete; conceptualization
Primary Actor	User level
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Looks up their own or another person's past match history Riot Games - Provides list of champions and items.
Preconditions	User has navigated to the site and the site is currently operational.
Success Post Condition	Five randomly selected champions with items are displayed.
Failed Post Condition	The web pages displays an error message

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario
1	User goes to the website application
2	The user then clicks the "Ultimate Bravery" button

3	The user then clicks the “Generate” button
4	The backend will perform an API call to retrieve a list of champion and items.
5	The back end will then randomly select five champions with six items for each.
6	The front end will display randomly selected champions
7	The user will see the information and may play the randomly chosen champions for the next game.
8	The user can decide to navigate to somewhere else on the site or leave.

EXTENSIONS or Alternate Flows

Step	Branching Action
5b	API Call fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.
7b	The user doesn’t like randomly chosen champions
	7b1. The user clicks the “Generate” button. 7b2. Repeat step 4 - 7.

SPECIAL REQUIREMENTS

Req Num	Requirement
1	Images of items and champions will be displayed

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	Variation
1	Should only show items that are available to purchase on the “Summoner’s Rift” map.

FREQUENCY OF OCCURRENCE: LOW

OTHER ISSUES

Issue Num	Issue
1	Some champion have special restrictions on items that they can have.
2	Possibly have it so that a boots item is guaranteed on the randomized selection.

Use Case # 15: Item Information

GENERAL CHARACTERISTICS	
Author	Gil Turner, Sung Kim, Irtiza Delwar
Last Update:	9/24/2016
Scope	Web Page
Level	User-level
Status	Incomplete, conceptualization
Primary Actor	User of the website.
Secondary Actors	Riot Games: Database
Stakeholders and Interests	User - Viewing statistics based on champion Riot Games - API provides statistics and data about each champion
Preconditions	N/A
Success Post Condition	The user is able to successful see accurate information about the desired item.
Failed Post Condition	The search for the item either fails or provides inaccurate information

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario
1	The user will navigate to the item information section of the website and click the item info button.

2	The user would then scroll through a menu of all the items and chose one.
3	An API call would be triggered and gather data about the selected item.
4	The website will display crucial information about the item such as cost, build path, and stats.
5	The user will then be able to look at the item information.
6	The user can then select another item or navigate to another portion of the website.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
4b	API call fails 1. Unable to access the API 2. Website display message saying the API is unavailable at the moment.
5a	More detailed item information
	4a1: By default the item would only return the information about cost, build path, and stats. The user may want more information 4a2: The user then can click on the more information section below the current item information. 4a3: The website will then display more detailed information that the API call was able to retrieve (such as sell cost, description, etc).

SPECIAL REQUIREMENTS

Req Num	**Requirement**

1	The item search should display the picture of the item
2	The build path should display the pictures of the items that are apart of the build path.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
1	Should support item search based on map (Ex: Items on some maps have different build paths depending on the map.

FREQUENCY OF OCCURRENCE: MODERATE

OTHER ISSUES

Issue Num	**Issue**
1	Possibly have other recommend items that are similar to the item selected
2	Possibly have recommended champions to build the item on.

Use Case # 16: Frequently Asked Questions

GENERAL CHARACTERISTICS	
Author	Sung Ho Kim, Gil Turner, Irtiza Delwar
Last Update:	9/24/2016
Scope	Web Page
Level	User Level
Status	Incomplete, conceptualization.
Primary Actor	User
Secondary Actors	N/A
Stakeholders and Interests	User - Finds answers to the questions that they had
Preconditions	N/A
Success Post Condition	Information and data about the desired user has been successfully displayed.
Failed Post Condition	The server doesn’t respond.

MAIN SUCCESS SCENARIO (or basic flow)	
Step	**Action -** description in words of each step in success scenario
1	User goes to the website application
2	The user then clicks the “Frequently Asked Question” button

3	The website will display frequently asked questions and answers to them.
4	The user sees this information and then can decide to navigate to somewhere else on the site or leave.

EXTENSIONS or Alternate Flows

Step	**Branching Action**
5	The users want to ask a question that is not listed.
	5a. The user scrolls to the bottom of the page. 5b. The user sees the contact information (i.e. email). 5c. The user can ask a question through the email address provided.

SPECIAL REQUIREMENTS

Req Num	**Requirement**
1	Questions on “Frequently Asked Question” page should easily be modified/updated by the website host.

TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	**Variation**
1	The data, including questions and answers, should be maintained by the website host.

FREQUENCY OF OCCURRENCE: OCCASIONALLY

OTHER ISSUES

Issue Num	Issue
1	Possibly have an email address somewhere on the website so people with high level concerns or bug catchers can contact us directly.