Use Cases

for

Wagyr

Version 2.0

Prepared by Team 2

CS 411: Software Engineering

5/04/2016

1. Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

2. Use Case Identification

1.1. Use Case ID

Give each use case a unique integer sequence number identifier. Alternatively, use a hierarchical form: X.Y. Related use cases can be grouped in the hierarchy.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.1.1. Created By

Supply the name of the person who initially documented this use case.

1.1.2. Date Created

Enter the date on which the use case was initially documented.

1.1.3. Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.1.4. Date Last Updated

Enter the date on which the use case was most recently updated.

3. Use Case Definition

1.1. Actors

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.

1.2. Trigger

Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.

1.3. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

1.4. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

1.5. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

1.6. Normal Flow

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered "X.0", where "X" is the Use Case ID.

1.7. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form "X.Y", where "X" is the Use Case ID and Y is a sequence number for the alternative flow. For example, "5.3" would indicate the third alternative flow for use case number 5.

1.8. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form "X.Y.E.Z", where "X" is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, “E” indicates an exception, and “Z” is a sequence number for the exceptions. For example “5.0.E.2” would indicate the second exception for the normal flow for use case number 5.

1.9. Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

1.10. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

1.11. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

1.12. Business Rules

List any business rules that influence this use case.

1.13. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

1.14. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

1.15. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case List

ID	***Primary Actor***	***Use Case Title***
UC1	New User	Sign-up/Manage settings
UC2	Existing User	Create Bet
UC3	Existing User	Accept Bet
UC4	Existing User	View Live Bets

Use Case Template

Use Case ID:	UC1		
Use Case Name:	Sign-up/Manage settings		
Created By:	Wagyr Dev Team	Last Updated By:	Wagyr Dev Team
Date Created:	2/17/2016	Date Last Updated:	5/04/2016

Actors:	New User
Description:	To register a new user with Waygr
Trigger:	Advertisement
Preconditions:	Email address
Postconditions:	User account created with supplied credentials “Success!” message & User summary
Normal Flow:	1. Home page (public) 2. Click register 3. Complete form and submit 4. Validate form submission 5. Redirect to “success” / user summary page
Alternative Flows:	1. Home page (public) 2. Click register

	3. Complete form and submit 4. Error: invalid form fields 5. Fix errors and resubmit form 6. Validate form submission 7. Redirect to “success”/user summary page
Exceptions:	Password length; Need valid email address
Includes:	Existing user logging into Wagyr normally
Priority:	High priority
Frequency of Use:	1/per actor
Business Rules:	Need to create an account in order to use Waygr
Special Requirements:	N/A
Assumptions:	Actor reached Wagyr public home page
Notes and Issues:	TBD: verify user’s age

Use Case ID:	UC2		
Use Case Name:	Create Wagyr		
Created By:	Wagyr Dev Team	Last Updated By:	Wagyr Dev Team
Date Created:	3/17/2016	Date Last Updated:	5/04/2016

Actors:	Existing User

Description:	To allow a user to create a new wagyr
Trigger:	Event to bet on/desire to create wagyr
Preconditions:	Wagyr user
Postconditions:	Wagyr appears on "Your Wagyrs"
Normal Flow:	1. Home Page 2. Search for Team 3. View Team's Schedule 4. Select game 5. Redirected to "Create a Wagyr" 6. Specify opponent 7. Specify desired amount of Waygr 8. Specify associated teams 9. Submit 10. Redirected to home page
Alternative Flows:	N/A
Exceptions:	Unable to create new Wagyr event
Includes:	Existing user creates new event normally
Priority:	High priority
Frequency of Use:	1/per actor
Business Rules:	Need to specify amount and opposition in order to create event
Special Requirements:	N/A
Assumptions:	Actor created wagyr
Notes and Issues:	Need to ensure that chosen opponent wants to participate in wagyr

Use Case ID:	UC3		
Use Case Name:	Cash Out (Receive Payment)		
Created By:	Wagyr Dev Team	Last Updated By:	Wagyr Dev Team
Date Created:	3/17/2016	Date Last Updated:	5/04/2016

Actors:	Existing User
Description:	Allow wagyr winner to cash out
Trigger:	User won wagyr
Preconditions:	Wagyr user Won wagyr
Postconditions:	Redirected to “success” page User should see money transferred onto specified debit card Wagyr should change from “active” to “fulfilled”
Normal Flow:	1. View “Your Wagyrs” 2. See updated status of Wagyr (Won) 3. Select “Cash Out” 4. Redirected to Receive Payment Form 5. Complete and submit form 6. Redirected to Success Page
Alternative Flows:	N/A
Exceptions:	Invalid debit card number
Includes:	Existing user creates wagyr
Priority:	High priority
Frequency of Use:	1/per actor
Business Rules:	25% Transaction fee is taken from wagyr amount
Special Requirements:	Need debit card
Assumptions:	Actor accepted bet
Notes and Issues:	Allow users to receive money through bank account numbers as well

Use Case ID:	UC4		
Use Case Name:	View Active Wagyrs		
Created By:	Wagyr Dev Team	Last Updated By:	Wagyr Dev Team
Date Created:	3/17/2016	Date Last Updated:	5/04/2016

Actors:	Existing User
Description:	Allow user to view active wagyrs
Trigger:	Desire to view wagyrs
Preconditions:	Wagyr User
Postconditions:	List of active/fulfilled wagyrs
Normal Flow:	1. Home page 2. Select “Wagyrs” to View “Your Wagyrs” 3. See list of active vs. fulfilled wagyrs
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	Existing user views wagyrs normally
Priority:	High priority
Frequency of Use:	1/per actor
Business Rules:	N/A
Special Requirements:	N/A
Assumptions:	Actor can view wagyrs

Notes and Issues:	Should be redirected to this page instead of the home page after creating a wagyr

Revision History

Name	Date	Reason For Changes	Version
UC1	2/17/16	Created new UC1	1.0
UC2	3/17/16	Created new UC2	1.0
UC3	3/17/16	Created new UC3	1.0
UC4	3/17/16	Created new UC4	1.0
UC1	5/04/16	Updated UC1	2.0
UC2	5/04/16	Updated UC2	2.0
UC3	5/04/16	Updated UC3	2.0
UC4	5/04/16	Updated UC4	2.0