Luis Montes
Jacob Serafin
Julian Shak
Kalen Wood-Wardlow

GramBeast

https://github.com/louiemontes/CS386_GramBeast

D2.2 - Use Cases

CS386

23 February 2017

Spring 2017

Professor Marco Gerosa

GramBeast
Use-Case : Validate User Account

1 Brief Description

This use case describes the process of logging into our application

2 Actor Brief Descriptions

2.1 User: person accessing the web application

2.2 Database: user account information for web application

3 Preconditions

The user has an active network connection

The user has an active user account registered in our database

4 Basic Flow of Events

1. The use case begins when the user visits our web application
2. The user selects user account type (Student/Instructor)
3. The system prompts the user for user credentials (User ID and Password)
4. Account information is sent to Database as a query. The Database replies with a go/no go reply telling if the query is ok.
5. The use case ends.

5 Alternative Flows

5.1 Invalid User

If in step 4 of the basic flow the Database query, then

1. The case ends at a failure condition.
2. //create account

6 Key Scenarios

6.1 No response from Database

7 Post-conditions

7.1 Successful Completion

The user successfully logs in with a valid account and is directed to the main page of the application.

7.2 Failure Condition

The failed login is logged to the database

8 Special Requirements

The user must exist in the database

Use-Case : Linking Account to a Class

1 Brief Description

This use case deals with the process of connecting a user account to a group of accounts, in our case this would be a given course they are enrolled in. This use-case is given off the assumption that validation has been accomplished for a “student” and “instructor” and the student gets access to their course group by the instructor.

2 Actor Brief Descriptions

2.1 Student: after validation, if account is identified as “student”, menu options will opt user to include a “class code” to be able to participate in their given course’s activities and start taking care of his/her pet.

2.2 Instructor: after validation, if account is identified as “instructor”, lists of courses they lead will appear within their interface. Option to create a new course group is available and the option to see the progress of each course already started all falls within their scope.

2.3 Database: a database would be needed for each account to hold their data; such as each student's course progress given as the status of a pet, and the instructors list of course groups they created and access to the progress of each student.

3 Preconditions

-The student or instructor has an active network connection

-The student or instructor has an active user account registered in our database, if not then one can be created.

-Students are enrolled in courses with instructors that offer our software as a part of their course work

4 Basic Flow of Events

1. The use case begins when the user visits our web application
2. Upon validation, their unique main menu shows their given course groups.
3. The course group is added to the students’ list via a unique code given by an instructor, as well as the student account is added to the list of students within an instructors course group.
4. The use case ends.

5 Alternative Flows

5.1 Instructor has no course groups

If in step 7 of the basic flow the instructor has no course groups, then

3. A course group is created, producing a unique code
4. The use case resumes at step 8

6 Subflows

6.1 Student has course group added

1. course group is listed within their main menu

2. upon the click on the group, their pet is shown and work can begin.
3. option to remove course group can be chosen.

7 Key Scenarios

7.1 Instructor Main Menu Activity

1. If no courses shown, step 8 can be repeated until all their courses are added.
2. Given all courses are added, a share button can be used to send unique code to students who are enrolled in their course so they can add themselves on GramBeast to the given course group.

8 Post-conditions

8.1 This condition requires that students are added to the group. Once this is achieved, options to start work, start petkeeping, remove course group, and add another course can be accomplished.

9 Special Requirements

Users are assumed to be meeting in a physical classroom with one another, able to share updates and learn new material needed to complete tasks within GramBeast.

Use-Case : Feeding a virtual pet

1 Brief Description

This use case deals with the process of a user logging into feed his virtual pet through a process of correct answer selection on a series of questions pertaining to their target language. Upon completing with correct answers, the pet is considered fed for the day.

2 Actor Brief Descriptions

2.1 User: After a user has successfully logged in, they will be presented a box of questions to try solving. These questions have unlimited attempts, but upon sending a wrong answer, a new question will replace the one the user could not solve. The user should not see questions they solved already presented to them again, except in the rare classroom case where an instructor may choose repeated questions.

2.2 Database: a database would be needed for each account to hold their data; specifically the pet's current hunger level and whether or not the number of correct answers since last user login has satied the pet's hunger.

3 Preconditions

-The user has an active network connection

-The user has an active user account registered in our database, if not then one can be created.

-While this application is made with the classroom setting in mind, anyone can create an account and generate a pet that needs feeding.

-The user has a pet in our database, and if not, can make a new one. However, the number of dead pets will be increased for every pet they cannot feed due to its starvation.

4 Basic Flow of Events

1. This use case begins when the user visits our web application
2. Upon validation, the user has a window of a queue of language questions they need to solve. They can vary in difficulty, some questions being as easy as selecting the right answer from among answers, to being as difficult as having to type in the correctly conjugated word or translated sentence.
3. If the user gets the answer wrong, a new question will be presented from our database of questions and given that our database can be provided from the instructor if the instructor so chooses to include custom questions, the user will probably have a lot of questions to try answering until seeing repeats.
4. Once the user has answered enough questions for their animal to no longer be hungry, they can continue answering questions if they so choose, but correct answers will no longer improve their animal's hunger level.

5 Alternative Flows

5.1 No Pet

If a user logs in and, say they were a new account, a user is given a prompt to make a new pet.

5.2 Starved Pet

If a user logs in and there pet is dead from starvation, they may make a new pet, but his or her profile now has an increase of one in their “Dead Pets” attribute.

6 Key Scenarios

6.1 The user feeds his or her pet.

7 Post-conditions

7.1 Successful Completion

The user successfully completes enough questions for the pet’s hunger to be satisfied, the user can then log off knowing his or her pet has survived another day (and in real life a user may be closer to extra credit at the end of his or her semester provided their instructor has offered this for keeping a Language-Pet alive until finals, which every one of the six interviewed language professors was in favor of doing).

7.2 Failure Condition

The failed question is replaced with another question that hopefully the user can complete successfully, but if still not and the system has ran out of new questions, then the queue of questions returns their first question again and the cycle continues.

8 Special Requirements

The user must exist in the database, and they must have a living pet to feed in our database.

Use-Case: Creating your pet

1 Brief Description

This use case deals with the user creating his or her virtual pet and customizing it to their liking. Some of the customization features of creating your pet would be its name, species, height, weight, eye color, fur color, facial features, nature of pet, and their background which will be their habitat.

2 Actor Brief Descriptions

2.1 User

When the user first uses the software or their pet dies they will have the option to create a new pet. When they create their new pet they will be brought to their customization menu, which will feature everything the user needs to change the features of their pet. The user can either choose their own pets features or they can shuffle pet cosmetics for random results. Once the user has finished customizing they save the changes, and their pet will be created for them to feed.

2.2 Pet

The pet will offer many different features to customize it. Some of these features will include the pet's species, name of the pet, the height and width of the pet, the pet's eye and fur color, and other distinct features including facial and body modifications making each pet very unique. The pet will also be able to choose its habitat based on either the species original habitat or the user's choice of the habitat of their pet (like the forest, desert, jungle, etc).

3 Preconditions

- In order to create the pet the user needs to have a new account or their past pet must be dead.
- In order to create the pet the user needs to have a validated account, as in, can log in and our database send him or her, the client, their correct profile to generate a pet from.

4 Basic Flow of Events

1) The user will be prompted to create a new pet if the user is a new account or his or her pet has passed away.
2) User will enter the customization menu for editing their pet's appearance.
3) The user will have the option to choose their pet's features or choose a random appearance.
4) Once the user has chosen their pet's appearance the user will select a habitat for their pet to live in.
5) Once the user is content with their pet's background it will save the pet's data to a database so it can be later accessed by the user so they can see their pet.

5 Alternative Flow of Events

5.1 User exits before saving pet

If the user exits before saving their pet's appearance and they do a soft log out (not saving their pet but logging out) then the pet's current customization will be saved and the user will be prompted to finish their pet's customization when they log back in. If the user does a hard log out (exiting the browser entirely without logging out) the user's data will not be saved and the user will be prompted when they log back to recreate their pet from scratch.

6 Key Scenarios

6.1 The user will customize their pet and the pet will be created for them to use until it dies.

7 Post-Conditions

7.1 Completion of Creating Their Pet

Once the user is content with their pet's appearance and habitat, the pet will be created and saved for the user to use, feed, and do other features with their pet as we release new features. The user will need to continually feed their pet or it will starve to death and they will have to create a new pet. The user will have the option to do minor customizations on their pet but not a full overhaul of its design.

8 Special Requirements

The user needs to have a valid account for their pet to save to the database, and the pet will only last as long as they feed it.

Group Participation

Julian Shak: Document setup, Use Case: Validating User, 3 User stories.
Jacob Serafin:
Luis Montes: Use Case: Feeding The Pet, 4 User stories.
Kalen Wood-Wardlow: Use Case: Creating a Pet, 3 User stories.